CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Protein design using artificial intelligence
Authors	Marcu, Stefan-Bogdan
Publication date	2025
Original Citation	Marcu, S-B. 2025. Protein design using artificial intelligence. PhD Thesis, University College Cork.
Type of publication	Doctoral thesis
Rights	© 2025, Stefan-Bogdan Marcu. - https://creativecommons.org/licenses/by/4.0/
Download date	2026-10-26 02:55:26
Item downloaded from	https://hdl.handle.net/10468/17423

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

OLLSCOIL NA HÉIREANN, CORCAIGH
NATIONAL UNIVERSITY OF IRELAND, CORK



Protein Design Using Artificial Intelligence

Thesis presented by

Ştefan-Bogdan, Marcu

0000-0002-2508-8753

for the degree of

Doctor of Philosophy

University College Cork

School of Computer Science

Head of School: Prof. Utz Roedig
Supervisors: Dr. Sabin Tăbîrcă
Prof. Mark Tangney

2025

Contents

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Scope of work . . . 2
1.3 Contributions And Publications . . . 3
1.3.1 AlphaFold . . . 3
1.3.2 ChemFlow . . . 3
1.3.3 SilverSurfer . . . 3
1.3.4 PuzzleSolver . . . 4
1.3.5 PS-GO . . . 4
1.3.6 PROFASA . . . 4
1.3.7 ProteinFlow . . . 4
1.4 Structure of the Thesis . . . 5
1.4.1 Chapter 1: Introduction . . . 5
1.4.2 Chapter 2: Background and Related Work . . . 5
1.4.3 Chapter 3: Protein Folding, AlphaFold . . . 5
1.4.4 Chapter 4: Molecular Recognition, Puzzle Solver . . . 6
1.4.5 Chapter 5: Parametric Protein Design, SilverSurfer & PS-GO . . . 6
1.4.6 Chapter 6: Protein Platform, Profasa & ChemFlow . . . 6
1.4.7 Chapter 7: Conclusion and Future Directions . . . 6

2 Background and Related Work **7**
2.1 Bioinformatics . . . 7
2.1.1 Proteins . . . 7
2.1.2 Protein Structure . . . 10
X-Ray Crystallography . . . 10
Nuclear Magnetic Resonance Spectroscopy . . . 12
2.1.3 Molecular Recognition . . . 13
2.1.4 Drug Discovery . . . 14
High-throughput screening . . . 16
Fragment-based drug discovery . . . 16
2.1.5 Drug Development Pipeline . . . 17
Target Identification and Validation . . . 18
Hit Identification . . . 18
Hit to Lead . . . 19
Lead Optimisation . . . 20

Preclinical Testing . . . 21
Clinical Trials . . . 22
2.1.6 Computer Aided Drug Design . . . 23
2.1.7 Virtual Screening . . . 23
Ligand Based Virtual Screening . . . 24
Structure-Based Virtual Screening . . . 25
2.2 Machine Learning, Deep Learning and Large Language Models . . . 26
2.2.1 Genetic Algorithms . . . 26
GA Mechanism of function . . . 28
Genetic Operators . . . 30
2.2.2 Supervised Learning . . . 32
Perceptron . . . 38
2.2.3 Deep Neural Networks . . . 39
2.2.4 Multi Layered Perceptrons . . . 41
2.2.5 Gradient Descent . . . 43
2.2.6 CNN . . . 44
2.2.7 Autoencoders . . . 50
2.2.8 Transformer . . . 55
2.2.9 Large Language Models . . . 59
2.2.10 Molecular Representations . . . 62
SMILES . . . 63
Molecular Fingerprints . . . 63
Voxelised Image Representation . . . 64
Molecular Graphs . . . 65
Learned Representations . . . 65

3 Protein Folding, AlphaFold **67**
3.1 Background on Protein Folding . . . 67
3.1.1 Importance of protein Folding . . . 67
3.1.2 Complexity of protein Folding . . . 68
The Energy Function Problem . . . 69
The Sampling Problem . . . 69
3.1.3 CASP and PDB . . . 70
Critical Assessment of Structure Prediction (CASP) . . . 70
The Protein Data Bank . . . 70
3.1.4 Fold Prediction Tecniques . . . 71
Physics Simulations . . . 71
Fragment Assembly . . . 71
Machine Learning . . . 71
3.2 AlphaFold . . . 72
3.2.1 Achievements in the CASP competitions . . . 72
AlphaFold . . . 72

AlphaFold 2 . . . 72
3.2.2 Methodology . . . 74
AlphaFold Algorithm . . . 74
AlphaFold 2 Algorithm . . . 76
3.2.3 Improvments of AlphaFold 2 over AlphaFold . . . 79
3.3 Computational techniques and technological Innovations . . . 80
3.3.1 AlphaFold . . . 80
3.3.2 AlphaFold 2 . . . 81
3.4 Potential applications in drug discovery . . . 82
3.5 Technical Challenges . . . 83
3.5.1 AlphaFold Limitations . . . 83
3.5.2 AlphaFold 2 Limitations . . . 83
3.6 Future Directions . . . 84
3.7 Conclusion . . . 85

4 Molecular Recognition, Puzzle Solver 87
4.1 Introduction . . . 87
4.1.1 Connection with Protein Design . . . 87
4.1.2 Motivation on tackling the problem . . . 88
4.2 Theoretical Foundations: From Jigsaw Puzzles to Molecular Recognition 89
4.2.1 Description of the Jigsaw Puzzle Challenge . . . 89
4.2.2 Historical Approaches . . . 91
4.2.3 Training Data Requirements . . . 92
4.2.4 Bézier Curves . . . 92
4.3 Proposed Solution . . . 93
4.3.1 Jigsaw Puzzle Generator . . . 94
4.3.2 AI Solver . . . 95
4.4 Lessons from the AI-Solver . . . 98
4.4.1 Knoweledge Gained . . . 98
4.4.2 Model Limitations . . . 99
4.5 Future Directions In Molecular Recognition Modeling . . . 100
4.5.1 AlphaFold and Molecular Recogntion . . . 100
4.5.2 The Next Frontier: Molecular Recognition Design . . . 100
4.6 Conclusion . . . 101

5 Parametric Protein Design, Silver Surfer & PS-GO 103
5.1 Introduction . . . 103
5.1.1 Theoretical Background . . . 104
5.1.2 Parametric Protein Design . . . 104
5.1.3 Challenges posed by searching the protein space . . . 106
5.1.4 Beyond the Known Protein Space . . . 107
5.2 PS-GO: Protein Search . . . 108
5.2.1 Parametric search strategy . . . 109

Data Ingestion . . . 109
Parameter Calculation . . . 110
Data Integration . . . 110
Parametric Search . . . 110
5.3 SilverSurfer . . . 110
5.3.1 Genetic Algorithms Particularities . . . 111
Encoding . . . 111
Mutation and Crossover . . . 111
Fitness . . . 112
5.3.2 Implementation . . . 112
5.3.3 Protein Generation . . . 113
Initialisation and Iterative Search . . . 113
Genetic Operators and Evaluation . . . 113
5.3.4 Results . . . 117
Validating the properties . . . 117
Parametric Generation . . . 117
5.4 Comparative Analysis . . . 119
5.4.1 Unique Approaches of PS-GO . . . 119
5.4.2 Unique Approaches of SilverSurfer . . . 119
5.4.3 Benefits of PS-GO and SilverSurfer . . . 120
5.4.4 Synergy Between PS-GO and SilverSurfer . . . 121
5.5 Conclusion . . . 121

6 Protein Platform, Profasa & ChemFlow 123
6.1 Introduction . . . 123
6.1.1 The Value of Workflows . . . 124
6.1.2 Motivation . . . 125
6.1.3 Contributions . . . 126
6.2 PROFASA: Enhancing Protein Analysis and Design . . . 126
6.2.1 Architecture Design . . . 127
Front End . . . 128
Back End . . . 129
6.3 ChemFlow: Automating Bioinformatics Workflows . . . 130
6.3.1 Workflows . . . 131
6.3.2 LLM . . . 131
6.3.3 Architecture Design . . . 132
Front End . . . 133
Back End . . . 135
6.4 Impact On Protein Design . . . 136
6.4.1 PROFASA Results . . . 137
6.4.2 ChemFlow Response . . . 138
Frontend . . . 138

Backend . . . 138
Workflow Generation . . . 139
6.5 Technical Innovations - LLM prompt optimisation . . . 139
6.5.1 Prompt Style Transfer . . . 139
6.5.2 Prompt Rephrasing . . . 140
6.5.3 Prompt Iterative Improvement & Ranking . . . 141
6.5.4 Prompt Correcting . . . 142
6.6 Challenges and Future Directions . . . 143
6.6.1 Challenges . . . 143
6.6.2 Future Directions . . . 144
6.7 Conclusion . . . 144

7 Conclusion and Future Directions **147**
7.1 Summary of Key Findings . . . 147
7.2 Future Direction . . . 148
7.2.1 Protein Strcture Prediction . . . 148
7.2.2 Molecular Recognition . . . 148
7.2.3 Parametric Protein Design . . . 149
7.2.4 Protein Platform . . . 149
7.2.5 General Outlook . . . 149

Bibliography **151**

Declaration of Authorship

I, Stefan-Bogdan MARCU, declare that this thesis titled, "Protein Design Using Artificial Intelligence" and the work presented in it are my own. I confirm that:

- The work I am submitting is my own and has not been submitted for another degree, either at University College Cork or elsewhere.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- All external references and sources are clearly acknowledged and identified within the contents.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have read and understood the regulations of University College Cork concerning plagiarism and intellectual property.

Signed:

_____

Date:

_____

NATIONAL UNIVERSITY OF IRELAND, CORK

Abstract

School of Computer Science

Doctor of Philosophy

Protein Design Using Artificial Intelligence

by Stefan-Bogdan MARCU

The emergence of new and resistant strains of pathogens, coupled with the slow and expensive process of drug discovery, has created an urgent need for innovative approaches in pharmaceutical development. This thesis explores how machine learning can be leveraged to accelerate the discovery and design of new pharmaceuticals. Each chapter addresses different aspects of the drug design process, proposing software solutions aimed at enhancing the productivity of researchers in the field.

The contributions section of the thesis begins with the examination of the fundamental problem of protein folding, focusing on AlphaFold and its unprecedented accuracy in predicting protein structures, which rivals experimental methods. The chapter discusses both the innovations brought forth by these models and their technical limitations.

Following this, the thesis pivots to the issue of Molecular Recognition, where protein conformation plays a critical role. This challenge is explored through the Puzzle Solver project, which uses a jigsaw puzzle analogy to approach the molecular recognition problem.

The exploration of protein predictors led to identifying a key issue in pharmaceutical research: the need for reverse protein prediction, specifically parametric protein design. This is addressed through the SilverSurfer and PS-GO projects.

The final section of the thesis focuses on integrating cutting-edge tools and simplifying workflows for pharmaceutical researchers. The PROFASA platform offers a single point for protein design needs, while ChemFlow lowers the technical knowledge barrier to designing workflows. This allows researchers to more easily utilise state-of-the-art tools and focus on their area of expertise.

Acknowledgements

I am immensely grateful to a number of people whose support was crucial in the completion of this thesis.

First, I would like to extend my deepest gratitude to my supervisors, Dr. Sabin Tăbîrcă and Prof. Mark Tangney, for their invaluable guidance, patience, and expert advice throughout my research journey. Their mentorship was essential in my academic development and inspiring on a professional level.

I am also thankful to all the friends and colleagues from the university, to Isabel Ronan and Andrew Nash, with whom I've travelled abroad and had very engaging conversations; to Yanlin Mi, Deniz Mevlevioğlu, Tianshu Xu and Dr. Venkata Vamsi Bharadwaj for their stimulating discussions, collaborative research and friendly atmosphere.

I would like to express my gratitude to Prof. Utz Roedig, the Head of the Department, and all the administrative staff at the school, whose assistance and friendliness are unmatched and have been invaluable throughout my studies.

I must acknowledge Dr. Anca Vasilescu, one of my Transilvania University teachers, whose recommendation was fundamental in my pursuit of this PhD.

On a personal level, I would like to thank my girlfriend, Denisa Cega, for her understanding, love, and encouragement, which were crucial to my well-being and success.

Lastly, and most importantly, I am forever indebted to my parents. Their unwavering support, encouragement, and sacrifices have been the cornerstone of my achievements. They have always set a standard for excellence and provided an aspirational model that serves as a guiding landmark throughout my life. Their sacrifices and belief in my potential continue to inspire and drive me towards my goals. This thesis stands as a testament to their love and belief in my potential.

To all of you, I am eternally grateful.

List of Figures

2.1 The figure depicts the building components of a protein using different representations: a) Uses chemical notation to illustrate the formula of one amino acid, its bonds and the constituting groups. b) Displays the same amino acid formula with its member groups overlayed on top of a standard ball and stick representation. c) Demonstrates how multiple amino acids bind to create a polypeptide chain and displays the characteristic torsion angles ψ and φ of the composing amino acids. d) Shows the ball and stick representation for the human haemoglobin (`1GZX`) with a length of 141 amino acids. 9

2.2 The figure illustrates the four common levels of abstraction when working with protein structure. Primary Structure is concerned with the linear sequence of amino acids in the chain; Secondary Structure focuses on immediate structures (α-`helices` and β-`sheets`); The tertiary structure centres on the 3D structure of the protein; The quaternary structure revolves around the way multiple protein subunits form into a functional complex. 9

2.3 The figure depicts the general stages and processes constituting X-ray crystallography. The proteins are crystalised; diffraction patterns are obtained with the help of X-rays. The diffraction pattern is computed into an electron density map, which is used to generate an atomic model of the protein conformation. The obtained model is then iteratively used to better understand the diffraction patterns, leading to iteratively improving the accuracy of the atomic model. 11

2.4 The figure depicts the main stages of identifying the molecular configuration of a compound using NMR spectroscopy. An MNR spectroscope, including a magnet, RF transmitter and detector components, is used to measure the attenuation curve. This data is used to create a spectrum showing the chemical shifts in the molecule; these shifts give clues to the local chemical configurations, which give strong clues towards the structure of the compound. 12

2.5 The figure graphically illustrates 3 different models for molecular recognition. A represents the classical lock and key mechanism. B exemplifies the process known as induced fit. C presents the conformation selection model. 13

2.6 Visual depiction of the number of compounds employed at each step of a typical drug discovery process: Clinical Space, Virtual Libraries, High-Throughput Screening, Hit-to-Lead, Lead Optimisation, Clinical Trial. The figure is not accurately scaled as the base layers dominate the top layers by 10s of orders of magnitude. 15
2.7 Three of the common strategies for Fragment-based Drug Discovery. . 16
2.8 The two main Phases of the Drug Development Pipeline and their constituent stages. 17
2.9 Lead optimisation (DMTA) cycle. 20
2.10 Illustration of the process of protein-encoding - the process of translating the real-world protein into numbers that can be processed by a computer. 27
2.11 Flowchart depicting the general workflow for the classical implementation of GA. 28
2.12 Illustration of the single-point crossover genetic-operator. 30
2.13 Illustration of the single-point mutation genetic-operator. 31
2.14 Schematic of the functionality of a single perceptron with four inputs, $x_1, x_2, \ldots, x_4$, with their afferent weights $w_1, w_2, \ldots, w_4$, the bias b. Each input is multiplied by its corresponding weight, the bias is added, and the result is passed through an activation function in order to get the output y. 38
2.15 Layered illustration of the layout and connections of a 2 hidden layer MLP with 4 inputs, 3 perceptrons per hidden layer and one output. . . 42
2.16 The mathematical operation of Convolution. 45
2.17 Discreete convolution. 46
2.18 Sparse connectivity. One input unit x_3 is highlighted together with the output units s that are affected by it. In the case of sparse interactions (top), when s is formed by convolution with a kernel of width 3, only three outputs are affected by x_3. In the case of a Fully Connected Network (bottom), the connectivity is no longer sparse, and as a consequence, all the outputs are affected by x_3. 47
2.19 Visualisation of a random subset of the feature filters in a Convolutional Neural Network at the first layer. 48
2.20 Graphical representation of the properties of equivariance (left) and invariance (right). 49
2.21 Workflow illustration of the conceptual functionality of an autoencoder. 50
2.22 Undercomplete autoencoder. 51
2.23 Denoising autoencoder. 53
2.24 Variational Autoencoder. 54
2.25 Internal workflow diagram of the self-attention mechanism. 56
2.26 Graphical illustration of the Transformer model. 58

2.27 Different 2D plane projections of an unknown 3D object (cube in this instance). 60
2.28 Illustration of the mapping from chemical structure to SMILES. 63
2.29 Schematic overview of the architecture of a 3D CNN model for protein-encoding used in protein-ligand binding affinity prediction. 64
2.30 Diagram illustration of a ligand-based graph and setup of a Graph Convolutional Network for protein-ligand binding affinity prediction. 65
2.31 Autoencoder learned representation from SMILES notation. 65

3.1 Illustrative depiction of the process of denaturation (the process in which a protein looses its 3D conformation) and renaturation (a protein assembling itself back in the correct fold from the denatured state). 68
3.2 Example showcasing the conformation of haemoglobin (1HCO), with a length of 141 amino acids, is illustrated above. Left uses the ribbon representation, putting the accent on the 2nd order structure; Right uses the ball and stick representation, evidentiating the 1st order structure and its complexity. 69
3.3 The results of the CASP13 competition; the bar graph displays the performance of the competitors, and the table underneath showcases the score of the top performers. The AlphaFold team competed under the name A7D with the group code 043. Source: predictioncenter.org results page [44]. 73
3.4 The results of the CASP14 competition; the bar graph displays the performance of the competitors, and the table underneath showcases the score of the top performers. The AlphaFold team competed under the name AlphaFold 2 with the group code 427. Source: predictioncenter.org results page [44]. 73
3.5 Component Workflow within the AlphaFold solution architecture. . . 74
3.6 Visual depiction of the histogram used within AlphaFold. Source: Jumper et al. [144]. 75
3.7 ResNet model architecture. Source: Jumper et al. [144]. 75
3.8 AlphaFold 2 Model architecture. Source: Yang et al. [325]. 77
3.9 Evoformer Module Architecture. Source: Yang et al. [325]. 78
3.10 Structure Module Architecture. Source: Yang et al. [325]. 79

4.1 Images of the 2JEL [237] protein, visualised with the help of MOL [273] on the RCSB website [230] reveal the similarity between the jigsaw puzzle problem and Protein-Protein interaction. A. - the Gaussian surface representation. B. - the x-ray representation. C. - Gaussian surface intersected with a clipping plane. 90
4.2 Representatives of each of the 18 classical types of jigsaw puzzle pieces. 90
4.3 Example of a bezier curve with 4 control points. 93

4.4 A. - Puzzle Pictorial and its selected background texture, B. - Cutout template generated with the help of the puzzle generator, C. - jigsaw puzzle scrambled over the background texture, D. - Correctly assembled jigsaw puzzle. . . . 94
4.5 Results of the encoder-decoder training; Green (false negatives), Red (false positives), Yellow (true positives). . . . 96
4.6 Puzzle solver architecture. . . . 97
4.7 Examples of the solvers' connection predictions (blue) over true connections (green). . . . 97

5.1 PS-GO protein search page; A. - Search component, B. - Results list component. Source: Mi et al. [200]. . . . 108
5.2 Exploration of the Monoisotopic Mass protein space. . . . 115
5.3 Exploration of the Isoelectric Point protein space. . . . 115
5.4 Exploration of the Instability Index protein space. . . . 116
5.5 Exploration of the GRAVY protein space. . . . 116
5.6 Illustrative example of the search path undertaken by the SilverSurfer in the complete protein space of proteins with a length of 4 amino-acids. 118

6.1 PROFASA combines all the major categories of tools, providing a single platform interface for all computational protein works. Source: Mi et al. [199]. . . . 126
6.2 Workflow example for a typical computational drug development process. . . . 130
6.3 ChemFlow Architecture. . . . 132
6.4 Snapshot of the Workflow Design GUI of ChemFlow. . . . 133
6.5 Snapshot of the Workflow Visualisation of ChemFlow. Green - Nodes that have finished processing; Yellow - Nodes currently being processed; White - Nodes yet to be processed. . . . 134
6.6 Visual representation of the 4 optimisation concepts employed by ChemFlow; a) Prompt Style Transfer, b) Prompt Rephrasing, c) Prompt Iterative Improvement and Ranking, d) Propmt Correcting. . . . 140
6.7 Visual depiction of the conceptual inspiration for the Prompt Rephrasing strategy. a) Yellow circle denotes the correct answer; Green dot denotes the starting point with the initial prompt; b) Blue dots indicate starting points obtained from prompt rephrasing; c) Blue dots after finishing the prompt evaluation. . . . 141

List of Abbreviations

AI	**A**rtificial **I**ntelligence
NMR	**N**uclear **M**agnetic **R**esonance
FDA	**F**ood and **D**rug **A**administration
HTS	**H**igh-**T**hroughput **S**creening
VS	**V**irtual **S**creening
LBVS	**L**igand-**B**ased **V**irtual **S**creening
SBVS	**S**tructure-**B**ased **V**irtual **S**creening
FBDD	**F**ragment-**B**ased **D**rug **D**iscovery
LBDD	**L**igand-**B**ased **D**rug **D**iscovery
SBDD	**S**tructure-**B**ased **D**rug **Discovery**
SAR	**S**tructure-**A**ctivity **R**elationship
QSAR	**Q**uantitative **S**tructure-**A**ctivity **R**elationship
ADME-TOX	**A**bsorption **D**istribution **M**etabolism **E**xcretion and **TOX**icity
TDD	**T**arget-driven **D**rug **D**iscovery
PDD	**P**henotypic **D**rug **D**iscovery
DMTA cycles	**D**esign, **M**ake, **T**est, **A**nalyse cycles
PK	**P**harmaco**kinetics**
CADD	**C**omputer **A**ided **D**rug **Design**
GA	**G**enetic **A**lgorithms
MSE	**M**ean **S**quare **E**rror
MAE	**M**ean **A**verage **E**rror
SVM	**S**upport **V**ector **M**achines
CNN	**C**onvolutionbal **N**eural **N**etworks
RNN	**R**ecurrent **N**eural **N**etworks
LSTM	**L**ong **S**hort-**T**erm **Memory**
GRU	**G**ayed **R**ecurrent **U**nits
MNIST	**M**odified **N**ational **I**nstitute of **S**tandards and **T**echnology
IMDb	**I**nternet **M**ovie **D**atabase
CPU	**C**entral **P**rocessing **U**nit
TPU	**T**ensor **P**rocessing **U**nit
GPU	**G**raphical **P**rocessing **U**nit
FFN	**F**eed**f**orward **N**etwork
MLP	**M**ulti-**l**ayered **p**erceptron
ReLU	**Re**ctified **L**inear **U**nit
ELU	**E**xponential **L**inear **U**nits

LLM	**L**arge **L**anguage **M**odels
MSA	**M**ulti-**S**equence **A**lignemnt
SMIELS	**S**implified **M**olecular-**I**nput **L**ine-**E**ntry **S**ystem
ECFP	**E**xtended **C**onnectivity **F**inger-**P**rints
PLEC	**P**rotein-**L**igand **E**xtended **C**onnectivity fingerprint
ECIF	**E**xtended **C**onnectivity **I**nteraction **F**eatures
RGB	**R**ed **G**reen **B**lue
CASP	**C**ritical **A**ssessment of **S**tructure **P**rediction
PDB	**P**rotein **D**ata **B**ank
GDT	**G**lobal **D**istance **T**est
GDT_TS	**G**lobal **D**istance **T**est **T**otal **S**core
FM	**F**ree **M**odelling
TBM	**T**emplate-**B**ased **M**odelling
IPA	**I**nvariant **P**oint **A**ttention
SOTA	**S**tate **O**f **T**he **A**rt
MRD	**M**olecular **R**ecognition **D**esign
mmCIF	**M**acro**m**olecular **C**rystallographic **I**nformation **F**ile
FASTA	**Fast A**ll
XML	E**x**tended **M**arkup **L**anguage
YAML	**Y**et **A**nother **M**arkup **L**anguage
SQL	**S**tructured **Q**uery **L**anguage
RMSD	**R**oot **M**ean **S**quare **D**ifference
IPC	**I**soelectric **P**oint **C**alculator
GRAVY	**Gr**and **A**verage H**y**dropathy
PS-GO	**P**rotein **S**tructure **GO**
PROFASA	**P**rotein **F**ragment **A**nd **S**tructure **A**nalysis
UX/UI	**U**ser E**x**perience / **U**ser **I**nterface
CORS	**C**ross-**O**rigin **R**esource **S**haring
ORM	**O**bject-**R**elational **M**apping
API	**A**pplication **P**rogramming **I**nterface

This thesis is dedicated to my supporting family, my caring parents, and my loving girlfriend; thank you so much for your unending love and support.

Chapter 1

Introduction

This chapter serves as the introduction to the thesis, describing the motivation behind the research, defining the scope of work and detailing the contributions made through various publications. The chapter concludes by providing a structured overview of the thesis layout.

1.1 Motivation

The unceasing battle against pathogens represents one of humanity's oldest and most relentless challenges. The emergence of new and resistant strains of bacteria, viruses and other pathogens necessitates the continuous development of novel drugs to ensure effective treatments and maintain public health [236]. The demand for new and innovative pharmaceuticals has never been greater, particularly as traditional drug discovery methods face speed, efficiency, and cost-effectiveness limitations [59].

In recent years, artificial intelligence has experienced a significant surge in both quality and performance. This technological boom offers unprecedented opportunities to revolutionise various fields, including drug discovery. By harnessing the power of machine learning, researchers can transform the drug development landscape, shifting from conventional searches for effective compounds to a targeted design approach.

AI-driven methodologies hold the promise to expedite the drug discovery process, making it faster and more efficient and reducing the prohibitive development costs [72, 132, 268]. These advancements may enhance the efficiency of identifying potential drug candidates and improve the precision in designing *de novo* drugs with specific desired functionality. This targeted approach holds the potential to develop therapies that are more effective and have fewer side effects, significantly contributing to personalised medicine.

The integration of AI in drug discovery stands to make significant contributions to advancing medical science and public health. By accelerating the discovery of new drugs and optimising existing ones, AI can play a central role in improving and potentially extending the quality of life for countless individuals. The potential

to substantially impact health outcomes adds a fulfilling dimension to the already intellectually stimulating challenge of protein design.

This research aims to address the pressing need for new and effective drugs to face evolving pathogens. By leveraging advancements in AI to improve the drug discovery process, it aims to contribute to the ongoing efforts to advance medical science and tackle one of humanity's oldest and most significant challenges.

1.2 Scope of work

The primary objective of this research is to explore and leverage artificial intelligence's capabilities within the domain of bioinformatics to enhance the drug discovery process. This investigation specifically focuses on the application of advanced machine-learning techniques to address both computational and biological challenges associated with the discovery and design of new pharmaceutical agents.

The scope of this research encompasses several key areas:

Predicting protein Shape through AI techniques: Focuses on capturing the modelling power of AI models and techniques in order to solve one of the classical questions in protein modelling: predicting the 3D conformational shape of a protein given its amino-acid sequence.

Machine learning for Protein-Ligand Interaction Prediction: This research aims to refine the predictive accuracy of protein-ligand binding models through the use of deep learning and other sophisticated machine learning methods.

Development of Machine Learning Enhanced Bioinformatics Tools: A significant part of the work involves developing and refining tools that integrate Machine learning to facilitate various aspects of drug discovery. These tools are designed to reduce the computational expertise required, making advanced bioinformatics analyses more accessible to the broader scientific community.

Optimisation of Drug Discovery Processes: The research aims to streamline the drug discovery pipeline by implementing AI-driven methodologies that can predict and model complex biological interactions more quickly and accurately than traditional methods. This includes automating aspects of the drug design process to enhance throughput and efficiency.

Ultimately, this research intends to demonstrate the transformative potential of integrating AI into bioinformatics, particularly in the context of drug discovery. The expected contributions of this work include the development of innovative computational tools and methodologies that enhance the efficiency and accuracy of the drug design process. By integrating AI techniques, this work seeks to facilitate more streamlined and effective approaches to pharmaceutical development, contributing valuable advancements to the field.

1.3 Contributions And Publications

This thesis presents several significant contributions to the field of protein design, reflected through publications and software developments that have each advanced the understanding and application of machine learning approaches in bioinformatics.

1.3.1 AlphaFold

The publication *An Overview of Alphafold's Breakthrough* [196], published in *Frontiers in Artificial Intelligence* examines the mechanisms behind AlphaFold and AlphaFold2, as well as their implications in the field of protein research. This mini-review explores the advanced methodologies employed by both versions of AlphaFold, highlighting the improvements from AlphaFold to AlphaFold 2, particularly focusing on their work mechanisms and the computational strategies that enable the prediction of protein structures. The review provides insights into how these technologies have reshaped protein structure prediction and discusses their potential applications in drug discovery and protein design, informing ongoing and future research in bioinformatics.

1.3.2 ChemFlow

ChemFlow, a software developed as part of this research, serves as a proof of concept for translating natural language descriptors into executable bioinformatics workflows with minimal technological knowledge required. This innovation significantly democratises access to complex computational tools, enabling a broader range of scientists to engage in sophisticated drug discovery efforts without requiring advanced programming knowledge. The associated paper, *ChemFlow: Bridging Computational Tools and Bioinformatics through LLM-Driven Workflow Automation* [193], accepted for publication in *Modelling and Development of Intelligent Systems*, outlines the development and application of ChemFlow; a proof of concept tool leveraging large language models to streamline the research processes in bioinformatics.

1.3.3 SilverSurfer

SilverSurfer represents another cornerstone of this thesis, a parametric protein engineering tool that utilises genetic algorithms to effectively explore and optimise the protein space. This tool's development was detailed in the paper *Silver Surfer: Navigating the Parametric Protein Space with Genetic Algorithms* [194], which received the Best Student Paper award at the conference on *Advances and Trends in Artificial Intelligence. Theory and Applications*. Silver Surfer provides researchers with an innovative tool for parametric protein design.

1.3.4 PuzzleSolver

PuzzleSolver introduces a dual-component tool comprising a realistic jigsaw puzzle generator and an AI-powered solver. This tool mimics the complexity of molecular recognition within a more controlled and quantifiable jigsaw puzzle environment. The development and implications of this novel approach are published in the paper *Generating Jigsaw Puzzles and an AI-Powered Solver* [195], in *Modelling and Development of Intelligent Systems*. This framework establishes a new approach to understanding and applying AI in solving complex, pattern-based problems analogous to those found in molecular biology.

1.3.5 PS-GO

PS-GO, a search engine developed within this research, specialises in parametric protein search, utilising pre-computed protein properties from ingested databases. This advanced approach allows users to search for proteins based on specific properties, making it highly effective for identifying proteins with desired characteristics. The tool's capabilities and its impact on protein research are detailed in the paper titled *PS-GO parametric protein search engine* [200], published in the *Computational and Structural Biotechnology Journal*. PS-GO represents a novel tool, offering researchers a powerful method for exploring proteins.

1.3.6 PROFASA

PROFASA stands as a pioneering platform designed for the detailed analysis of protein function and structures. This platform's capabilities were encapsulated in the paper *PROFASA—a web-based protein fragment and structure analysis workstation* [199], published in *Frontiers in Bioengineering and Biotechnology*. PROFASA offers an integrated environment for researchers to perform comprehensive protein analysis, enhancing the efficiency and depth of protein studies.

1.3.7 ProteinFlow

ProteinFlow stands as an innovative framework offering advanced tools for data processing, feature extraction and interpretation. It stands out by directly addressing the complexities of multidimensional protein data sets through parallel data processing strategies that demonstrate a significant improvement over traditional methods. Through the reduction of processing time and expanding the scope of biologically significant features identified, ProteinFlow emerges as an essential tool in effectively tackling the analysis of intricate protein property space. The framework's properties have been published in the paper *ProteinFlow: An advanced framework for feature engineering in protein data analysis* [201] part of the *Biotechnology and Bioengineering* journal.

Together, these publications and software tools represent a comprehensive effort to advance the field of protein design, contributing significantly to the broader goals of medical science and public health. Each contribution underscores the potential of advanced computational techniques in bioinformatics and establishes a foundation for future research in this vital area.

1.4 Structure of the Thesis

This thesis is organised into seven distinct chapters, each focusing on a different aspect of protein design, employing different machine learning techniques to empower the researcher. It builds a comprehensive narrative from foundational concepts to advanced applications and reflections on future directions:

1.4.1 Chapter 1: Introduction

Chapter 1 serves as the entry point into the thesis, setting the stage with an exploration of the motivation behind this research. It delves into the contributions made through publications and software tools that have emerged as a result of this work, providing a preview of the innovative approaches taken in the field of bioinformatics. This chapter ends by outlining the structure of the entire thesis.

1.4.2 Chapter 2: Background and Related Work

This chapter lays the groundwork by discussing the essential biological and computational concepts that underpin the thesis. The first section introduces key biological aspects, including proteins, their structures, the drug development process, and the role of computer-aided design in modern bioinformatics. The second section transitions into the realm of machine learning, covering genetic algorithms, the basics of supervised learning, and various staple architectures like Convolutional Neural Networks and Transformers, setting the stage for their application in subsequent chapters.

1.4.3 Chapter 3: Protein Folding, AlphaFold

Centering on the significant advancements in protein folding. Chapter 3 builds upon the work published in the "An Overview of Alphafold's Breakthrough" [196] paper. It discusses the critical role of protein folding in biological functions and how AlphaFold and AlphaFold2 have revolutionised this area. This chapter addresses the technical innovations and challenges of these methodologies.

1.4.4 Chapter 4: Molecular Recognition, Puzzle Solver

Expanding the focus to protein interactions, Chapter 4 leverages the analogies drawn from solving jigsaw puzzles, as discussed in the publication "Generating Jigsaw Puzzles and an AI-Powered Solver" [195]. This approach helps explore machine learning models for predicting protein interactions, offering insights into the complexities of molecular recognition.

1.4.5 Chapter 5: Parametric Protein Design, SilverSurfer & PS-GO

Chapter 5 explores the implementation of property prediction models and tools for parametric protein design, grounding in the works "PS-GO parametric protein search engine" [200] and "Silver Surfer: Navigating the Parametric Protein Space with Genetic Algorithms" [194]. It introduces the innovative tools PS-GO and Silver Surfer, designed to navigate and manipulate the protein space efficiently, showcasing the practical applications of the theoretical discussions from previous chapters.

1.4.6 Chapter 6: Protein Platform, Profasa & ChemFlow

Building on the technological advancements discussed in Chapter 5, Chapter 6 focuses on enhancing user experience and accessibility. It details the development of PROFASA and ChemFlow, platforms that integrate cutting-edge protein property predictors and workflow automation to lower the technical barriers for bioinformatics researchers, as highlighted in the papers "PROFASA—a web-based protein fragment and structure analysis workstation" [199] and "ChemFlow: Bridging Computational Tools and Bioinformatics through LLM-Driven Workflow Automation" [193].

1.4.7 Chapter 7: Conclusion and Future Directions

The final chapter synthesises the findings from the entire thesis, summarising the key results and contributions. It discusses the implications of this research in the broader context of medical science and public health and proposes promising avenues for future research, aiming to inspire ongoing innovation in the field of protein design and drug discovery.

Chapter 2

Background and Related Work

Chapter 2 lays the foundational knowledge for the thesis. It provides an extensive review of the literature through the exploration of theories and prior work in the fields of bioinformatics and machine learning. This chapter serves to contextualise the research within the broader scientific landscape and highlight advancements that have shaped the current state of protein structure prediction and computational drug design.

2.1 Bioinformatics

Bioinformatics is the interdisciplinary branch of molecular biology situated at the intersection of biology and computer science. It primarily focuses on the application of computational techniques to analyse large-scale biomolecular data, encompassing diverse subjects such as structural biology, genomics and gene expression studies [189]. A dynamic and rapidly evolving field of pivotal importance, bioinformatics is extensively applied across various domains, including drug production, gene therapy, agriculture, etc. [300]. Integrating knowledge from molecular biology, chemistry, mathematics, statistics and computer science [300], the study of bioinformatics provides powerful tools and methods for deciphering and understanding the complex functionality of biological processes and functions.

The role of bioinformatics has become of increasing importance due to the emergence of high-throughput technologies such as next-generation sequencing and mass spectrometry [175]. These technologies have enabled the generation of vast quantities of data at an unprecedented scale and speed, highlighting the necessity for sophisticated computational tools required to manage, analyse and interpret this wealth of information.

2.1.1 Proteins

Proteins have been an integral part of human knowledge long before the formal scientific understanding of their molecular structures. Historically, the practical applications of protein in everyday life are evident, ranging from food preparation and preservation to baking, brewing and traditional medicine. These early uses demonstrate an early empirical grasp of protein functionality. However, it was not until

the advancements in biochemistry during the 20th century that a profound transformation occurred in the human understanding of proteins, brought by the progress in deciphering the intricate details of protein sequences and their three-dimensional structures.

Their involvement in many of the essential and culture-defining historical human activities marks the historical indirect study of proteins through baking and brewing as one of the earliest instances of human engagement with biochemistry. The relatively modern discovery of the involvement of proteins in virtually all of the biological processes [2] rightfully designates them as one of the main subjects of study within the field of bioinformatics. Proteins are complex macromolecules that are critical for the structure and function of all living organisms. They fulfil diverse roles, such as catalyzing metabolic reactions, replicating DNA, responding to stimuli, and transporting molecules across cellular boundaries. Proteins are integral to the cellular infrastructure, working in physical or functional association with other proteins or biomolecules [2]. Understanding these dynamic and multifaceted molecules is of crucial importance for the development of new and improved pharmaceuticals.

Proteins are abundantly present in all living organisms. For instance, *Schizosaccharomyces pombe*, a type of fission yeast commonly used in traditional brewing and later adopted as a model organism for molecular and cell biology, contains approximately 60 million protein molecules with expression levels ranging from a few copies to as many as 1.1 million copies for some proteins [2]. Reflecting their critical role in cellular structure and function, proteins constitute about 50% of the dry mass of a cell across various species [2]. This variability and prevalence highlight the crucial role proteins play across different fields of biology.

Proteins are fundamentally composed of one or more polypeptide chains, which are sequences of amino acids (depicted in Fig. 2.1 a, b) linked together by the covalent bonds. At the core of these chains lies the central protein structure, also known as the protein backbone, which is characterised by a repeating sequence of nitrogen, alpha carbon, and carbonyl carbon (Fig. 2.1 c). Each amino acid within the polypeptide chain shares a common substructure centred around an alpha carbon. The α-carbon is covalently bonded to an amino (NH_2) group (green box in Fig. 2.1 a,b), a carboxyl ($COOH$) group (blue box in Fig. 2.1 a,b) and a hydrogen atom. The distinct identity and physicochemical properties of each amino acid are determined by its side chains, known as R groups (yellow box in Fig. 2.1 a,b). These R groups vary widely among the different amino acids, influencing the structure, functionality, and interaction capabilities of the protein. The specific sequence and composition of the R groups define the protein's overall structure (Fig. 2.1 d) and its biological role.

The structural organisation of proteins is conventionally described at four distinct levels of abstraction, each capturing different aspects of their complex structure and functionality. The first level of abstraction is the primary structure; it refers to the linear sequence of amino acids in the polypeptide chain (Fig. 2.2 a). At the

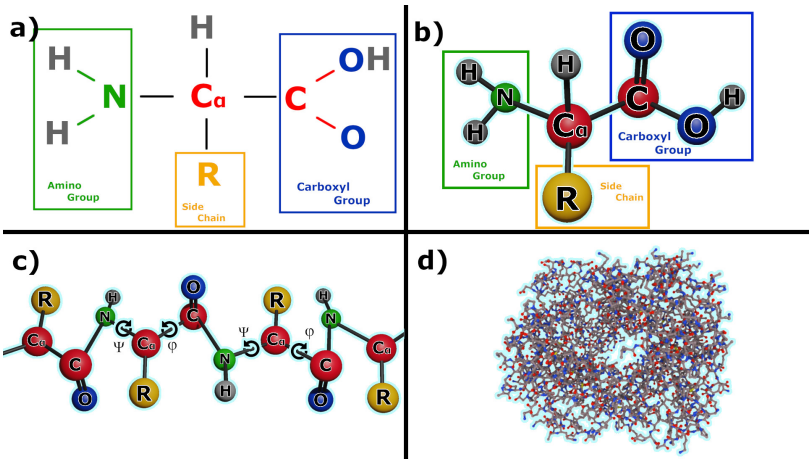


FIGURE 2.1: The figure depicts the building components of a protein using different representations: a) Uses chemical notation to illustrate the formula of one amino acid, its bonds and the constituting groups. b) Displays the same amino acid formula with its member groups overlayed on top of a standard ball and stick representation. c) Demonstrates how multiple amino acids bind to create a polypeptide chain and displays the characteristic torsion angles ψ and φ of the composing amino acids. d) Shows the ball and stick representation for the human haemoglobin (`1GZX`) with a length of 141 amino acids.

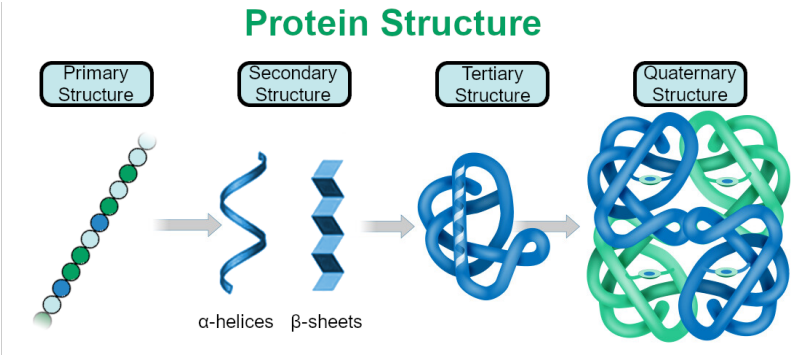


FIGURE 2.2: The figure illustrates the four common levels of abstraction when working with protein structure. Primary Structure is concerned with the linear sequence of amino acids in the chain; Secondary Structure focuses on immediate structures (α`-helices` and β`-sheets`); The tertiary structure centres on the 3D structure of the protein; The quaternary structure revolves around the way multiple protein subunits form into a functional complex.

second level, we encounter the secondary structures, which are characterised by patterns of hydrogen bonding between peptide backbone atoms forming alpha-helixes and beta-sheets. The alpha-helix is a right-handed coiled strand, whereas the beta-sheet consists of beta strands connected laterally by at least two or three backbone hydrogen bonds (Fig. 2.2 b). At the third level, we encounter the tertiary structure, which concerns itself with the three-dimensional shape of the entire polypeptide, including the spatial arrangement of the secondary structure. The shape is influenced by various interactions such as hydrophobic interactions, hydrogen bonds, disulfide

bonds, and ionic bonding between the side chains of the amino acids. These interactions cause the protein to fold into a specific three-dimensional structure that is biologically active (Fig. 2.2 c). The fourth and highest level of structural organisation is the quaternary structure. This level involves the assembly of multiple protein subunits into a functional complex (Fig. 2.2 d). Together, these four levels of protein structure provide a framework for understanding the complex dynamics of protein interactions.

The intrinsic link between a protein's biological function and its three-dimensional, native structure, is foundational to the understanding of its biological activity. The sequence of amino acids fundamentally encodes the unique three-dimensional structure of each protein [138]. This specific configuration is essential; without it, proteins may lose their functionality or even develop detrimental configurations [274]. The folding process, in which proteins transform from a linear chain of amino acids into a complex, folded structure, highlights how delicate the balance between structure and function is. Even seemingly minor alterations in the protein's sequence can lead to significant changes in its behaviour. This concept lays the foundation for the field of protein function prediction, which focuses on the study of amino acid sequences, structural configurations, and biomolecular interactions to predict the functional capabilities of the protein. The inverse problem, the process of engineering new proteins with desired functionality by optimising amino acid sequences to form specific structural attributes, is the focus of the field of protein design.

2.1.2 Protein Structure

The structure of a protein dictates its biological function, making the determination of its three-dimensional atomic configuration crucial for biological research and pharmaceutical development. There are two common modern techniques for experimentally determining the structure of a protein at the atomic level: X-ray crystallography and Nuclear Magnetic Resonance (NMR) Spectroscopy.

X-Ray Crystallography

X-ray crystallography stands as a foundational technique in structural biology, playing a pivotal role in the elucidation of the atomic structure of macromolecules. The inception of this method is anchored in the early suspicions of Max von Laue, who hypothesised that the wavelength of X-rays discovered by Wilhelm Conrad Röntgen in 1895 might be comparable to interatomic distances within crystals [97]. This led him to experiment with a blue crystal of copper sulfate pentahydrate ($CuSO_4 \cdot 5H_2O$), yielding diffraction images. Max Von Laue formulated a physical explanation of the observed diffraction images [97]. This experiment set the stage for further developments by Sir William Henry Bragg and his son, Sir William Lawrence Bragg, who recognised the utility of diffraction in investigating crystal structures. Notably, Sir William Lawrence Bragg formulated an elegant mathematical relation known as

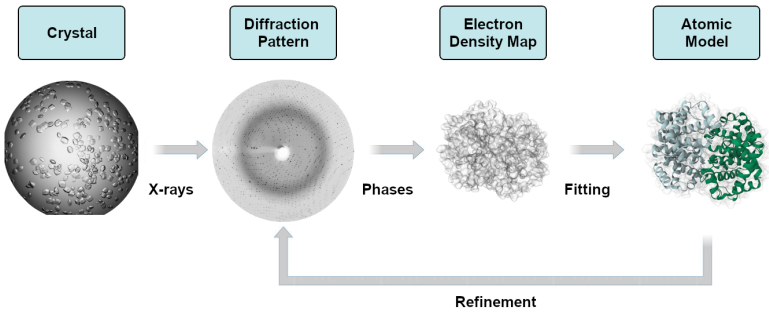


FIGURE 2.3: The figure depicts the general stages and processes constituting X-ray crystallography. The proteins are crystalised; diffraction patterns are obtained with the help of X-rays. The diffraction pattern is computed into an electron density map, which is used to generate an atomic model of the protein conformation. The obtained model is then iteratively used to better understand the diffraction patterns, leading to iteratively improving the accuracy of the atomic model.

Bragg's Law, $n\lambda = 2d \sin\Theta$, which describes the relationship between the angles of diffraction (Θ), the wavelength of the X-rays (λ), and the interplanar spacings (d) withing the crystal lattice [33].

In order to experimentally determine the structure of a macromolecule, X-ray crystallography involves several key steps. Initially, a sufficient quantity of the macromolecule must be produced in a pure form. In the next step, researchers explore various conditions under which the macromolecule can crystallise, followed by the crystal production in which the crystals of the macromolecule are grown, obtaining a crystalised protein similar to the Crystal depicted in Fig. 2.3. To ensure accurate diffraction patterns, these crystals must be of high quality, with regular lattice structures. The preparatory steps are followed by the data collection step. In this step, the crystals are subjected to X-ray diffraction, where X-rays are shone upon the crystal, creating diffraction patterns influenced by the atoms within the crystal similar to the Diffraction Pattern illustrated in Fig. 2.3. These patterns provide information about the amplitudes of the diffracted waves. Notably this step does not provide any phase information. The phase problem-solving step tackles the challenge of obtaining the phase information through methods like multi-wavelength anomalous diffraction, single-wavelength anomalous diffraction, multiple isomorphous replacements or molecular replacement techniques. Once the data is collected from the experiments, the final step is constituted by computing the electron density map from the collected data (an example of an Electron Density Map can be observed in Fig. 2.3). This map is then used to build and refine the macromolecular structure (depicted under Atomic Model in Fig. 2.3), considering the known covalent bond geometries and other structural data [26].

One of the key strengths of X-ray crystallography lies in its versatility. The technique can be used to experimentally determine the structure regardless of the size and complexity of the macromolecule, provided that it can be crystallised. Most

globular macromolecules meet this criterion [26], although the crystallisation process may slightly alter their local structures.

Nuclear Magnetic Resonance Spectroscopy

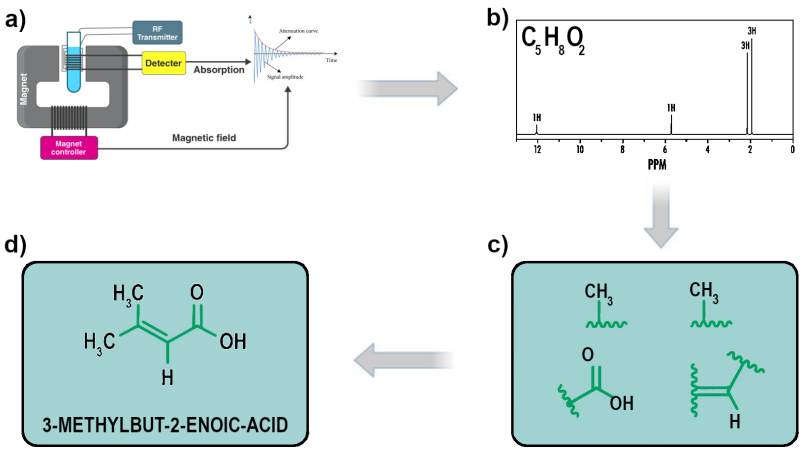


FIGURE 2.4: The figure depicts the main stages of identifying the molecular configuration of a compound using NMR spectroscopy. An MNR spectroscope, including a magnet, RF transmitter and detector components, is used to measure the attenuation curve. This data is used to create a spectrum showing the chemical shifts in the molecule; these shifts give clues to the local chemical configurations, which give strong clues towards the structure of the compound.

Nuclear magnetic resonance (NMR) spectroscopy is another bedrock technology in structural biology used to determine the three-dimensional structure of macromolecules at the atomic level. The phenomenon of nuclear magnetic resonance was discovered by Isidor Isaac Rabi, for which he received the Nobel Prize in Physics in 1944 [293]. In 1946, Purcell and Bloch independently developed NMR spectroscopy [205], a significant advancement for which they were awarded the Nobel Prize in Physics in 1952 [294]. This technique allows scientists to figure out the relative positions of various atomic elements through their responses to magnetic pulses [128]. The resulting constraints are then used to determine the actual three-dimensional structure of the macromolecule.

The process of NMR spectroscopy begins with the preparation of a macromolecule sample labelled with specific isotopes, usually ^{13}C and ^{15}N, to enhance the sensitivity and resolution of the NMR measurements. The next step involves collecting data (Fig. 2.4 a) and assigning the chemical shifts of hydrogen (H), nitrogen (N), and carbon (C) (Fig. 2.4 b) atoms in the macromolecule. Using the distance and/or orientation restraints derived from specific NMR experiments (Fig. 2.4), researchers then calculate and refine the structure of the macromolecule (Fig. 2.4 d).

One of the main advantages of NMR spectroscopy is that it can be carried out

in a concentrated solution rather than in a crystal form, allowing the study of proteins in conditions that more closely mimic their natural environment [26]. Additionally, NMR spectroscopy can define certain dynamic properties of macromolecules, providing insights into their flexibility and conformational changes [26]. However, this technique is generally limited to macromolecules with molecular weights of less than 30 kDa, due to the complexity and overlapping signals in larger molecules [26].

2.1.3 Molecular Recognition

Molecular Recognition, also known under the name of protein docking, is the fundamental physical principle underlying protein-ligand interactions, where specific interactions occur between two or more molecules through noncovalent bonds. These interactions are driven by a combination of intermolecular forces, including Van der Waals forces, electrostatic interactions, and hydrogen bonding. The precise nature of these noncovalent bonds ensures that molecules fit together in a lock-and-key manner, allowing for the high specificity and affinity necessary for biological function.

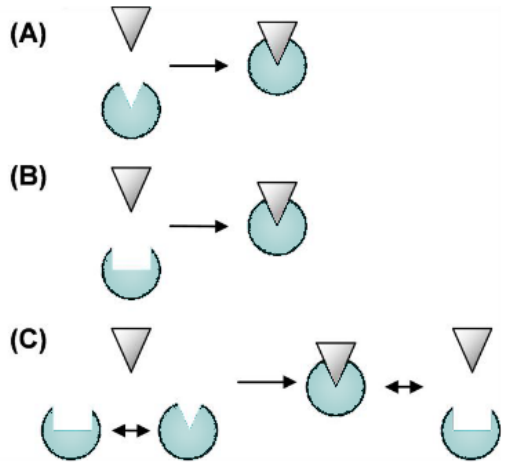


FIGURE 2.5: The figure graphically illustrates 3 different models for molecular recognition. A represents the classical lock and key mechanism. B exemplifies the process known as induced fit. C presents the conformation selection model.

The lock and key mechanism, introduced by Emil Fischer in 1894 [94] (Fig. 2.5 a), is one of the most well-known models for molecular recognition. According to this model, the active site of a host molecule possesses a specific, fixed shape. Much like a key fitting into a lock, a guest molecule will only bind to the active site if its shape and chemical properties exactly complement those of the active site. One notable deficiency of this model is constituted by the lack of consideration of the inherent flexibility of biological molecules, which can undergo conformational changes to facilitate binding.

The induced fit model, proposed by Koshland in 1958 [152] (Fig. 2.5 b), hypothesises that the binding of a ligand to a protein induces conformational changes in

the protein structure, enabling initially incompatible molecules to adopt conformations more conducive to binding. This dynamic adaptation contrasts with the static nature of the lock-and-key model. Despite its conceptual advantages in explaining molecular recognition, the induced fit model is computationally intensive, forcing practical applications to restrict the flexibility property to a limited number of side chains around the active site in order to reduce the computational power required to simulate conformational changes.

The conformation selection models, supported by Monod et al. [208] and Rubin et al. [255] (Fig. 2.5 c), postulates that proteins and ligands exist in a dynamic equilibrium of multiple conformations independently. According to this model, molecular recognition occurs when a ligand binds to a protein conformation that is already complementary without requiring significant conformational adjustments post-binding. This implies that the binding event selects the pre-existing conformation (that best fits the ligand) rather than inducing a fit, as proposed by the induced fit model. Simulating the molecular recognition against an ensemble of conformations is computationally expensive due to the necessity of running full simulations for each possible protein conformation, often requiring the use of dynamic simulations [13].

Despite significant advances in understanding the mechanism for molecular recognition, the debate over which model, induced fit or conformation selection, more accurately describes the process remains unresolved [47, 104]. Understanding these mechanisms is crucial for advancing our ability to design and predict protein-ligand interactions, which is one of the essential mechanisms that is situated at the foundation of drug discovery.

2.1.4 Drug Discovery

Drugs are an essential building block of a functioning health system, serving as crucial instruments in the treatment, prevention, and management of various diseases. Defined as chemical or biological entities, drugs modulate the physiological state of the body. They achieve this modulation by interacting with one or more target biomolecules, often proteins, to alter their functionality and restore or maintain health. The study and design of these interactions are central to the study of bioinformatics, which provides computational tools and methods to understand and predict the complex dynamics between drugs and their protein targets. The drugs' ability to influence biological pathways underscores their critical role in medical science and public health.

Traditionally, in 20th-century drug discovery, phenotypic screening approaches were used to identify promising drug candidates [207]. A notable example is the discovery of Penicillin in 1928 by Alexander Fleming [102], who observed that certain compounds present in mould inhibited bacterial growth, without understanding the underlying mechanism. Despite its historical success, phenotypic screening faces

significant challenges, particularly in hit validation and target deconvolution. Consequently, modern drug discovery methods favour a single-target approach, which has become widely adopted. However, recent studies have demonstrated that many drugs initially believed to be specific single-target agents actually derive their efficacy from polypharmacology, acting on multiple drug targets [30, 227].

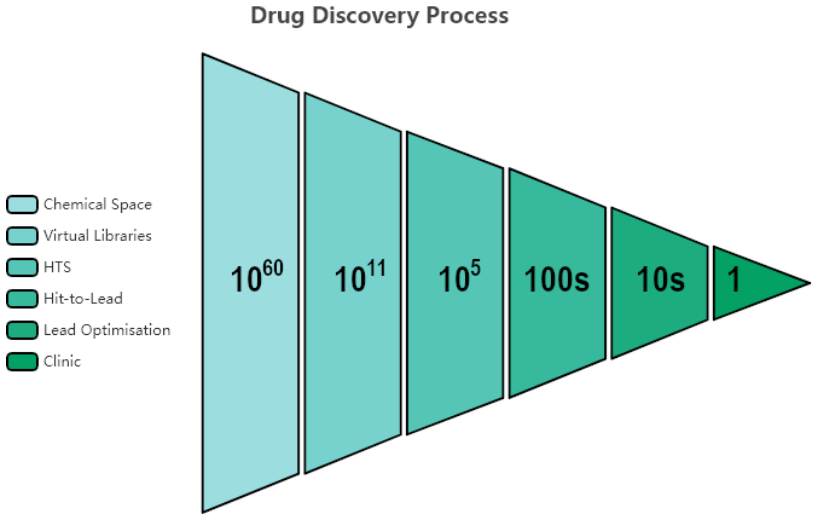


FIGURE 2.6: Visual depiction of the number of compounds employed at each step of a typical drug discovery process: Clinical Space, Virtual Libraries, High-Throughput Screening, Hit-to-Lead, Lead Optimisation, Clinical Trial. The figure is not accurately scaled as the base layers dominate the top layers by 10s of orders of magnitude.

Drug discovery is an extremely challenging, multi-stage process that includes a wide array of research areas, including biology, chemistry, pharmacology, and clinical sciences. This complexity is reflected in the high costs associated with drug development, estimated to range between 2-3 billion dollars per drug [72, 266, 268]. A significant portion of these costs arises from the high chance of failure, particularly due to high late-stage attrition (see Fig. 2.6). The drug development process is notoriously time-consuming, often taking over 10 years from initial discovery to market approval [228, 266]. Despite extensive efforts, on average, only 12.9 new molecular entities are approved by the FDA every year [89]. The need to develop drugs efficiently, effectively and in a timely manner has been demonstrated by the global emergency caused by the coronavirus pandemic.

The two main avenues for drug discovery are high-throughput screening (HTS) and fragment-based drug discovery (FBDD). HTS involves rapidly testing thousands to millions of compounds against biological targets to identify active compounds. FBDD focuses on identifying smaller chemical fragments that bind to the target, which are then optimised into drug candidates.

High-throughput screening

High-throughput screening (HTS) is a cornerstone of modern hit discovery employed by all major pharmaceutical companies [26]. This method has traditionally been utilised to search for active compounds in a single-target approach, yielding potential inhibitors that demonstrate biological activity. Extremely large libraries of in-house compounds, alongside public databases (Fig. 2.6), are routinely tested by the major pharmaceutical players for activity against target biomolecules or closely related structures.

HTS can be conducted through two primary methods: virtual screening and physical assays. Virtual screening is a computational method employed in the prediction of biological activity that will be elaborated further in the Computer-Aided Drug Design section. Physical assays are extensive experiments designed to measure specific biological processes or molecules. These assays typically involve evaluating enzyme activity, cell viability or gene expression. Many of the physical assays rely on radioactivity or fluorescence-based approaches [26]. They are typically performed in 384-well plates with 20-µL volumes to mitigate the cost of conducting numerous assays [26].

Despite its advantages, HTS has its limitations. One significant limitation is the overabundance of hit compounds, which exacerbates the problem of false positives. Even with a false positive rate as low as 0.01%, testing a billion compounds would result in 10,000 false positives. One approach to limit the number of false positives relies on identifying compounds that are potential Pan Assay Interference Compounds (PAINS).

Another limitation of HTS is that it is resource-intensive and often results in hits having a very small probability of binding, resulting in hits that are difficult to optimise [163, 246].

Fragment-based drug discovery

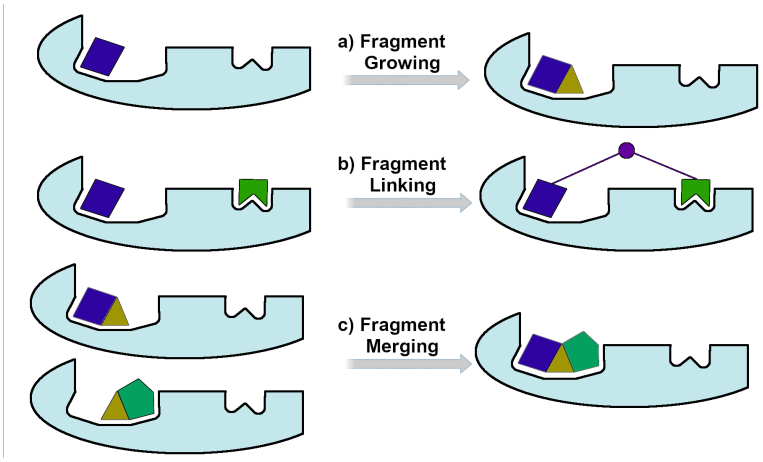


FIGURE 2.7: Three of the common strategies for Fragment-based Drug Discovery.

Fragment-based drug discovery (FBDD) is one of the mainstream methods used in modern drug discovery, with over 30 fragment-derived drug candidates in clinical trials and two fragment-derived drug approvals in 2016 [86] and growing to almost 40 new molecular entities approved in 2023 [15]. This method has greatly benefitted from recent advances in structural biology, which have enhanced its effectiveness and application [176].

FBDD employs smaller-than-drug-like compounds, known as fragments, to identify low-potency but high-quality leads. These fragments are typically less complex than traditional drug-like molecules. The primary advantage of using fragments lies in their ability to bind weakly but specifically to different regions within the target's binding pocket (Fig. 2.7).

The development process in FBDD involves assembling fragments based on their fit within the target pocket. There are three main strategies for drug development: growing, linking and merging [272]. In the growing strategy, fragments that bind weakly to the target are elaborated upon to increase their size and improve their binding affinity (Fig. 2.7 a). The linking strategy involves identifying two or more fragments that bind to adjacent sites within the target pocket and chemically linking them to form a single, more potent molecule (Fig. 2.7 c). The merging strategy combines overlapping fragments into a new compound that retains the desirable features of each original fragment (Fig. 2.7 d).

One of the disadvantages of FBDD is its reliance on structural data to guide the design and optimisation of fragment hits. Structural biology techniques, such as X-ray crystallography and NMR spectroscopy, are crucial in determining how fragments interact with their target proteins. This structural information is what enables the researchers to refine and optimise fragments into lead compounds with high affinity and specificity [214].

Despite its limitation on available structural data, FBDD is commonly employed in the drug discovery pipeline due to its higher hit rates compared to HTS. Hit rates in FBDD can be 10-1000 times higher than those achieved with HTS [114]. This increased efficiency is partly due to the smaller size and simplicity of the fragments, which allows for a more comprehensive sampling of the chemical space.

2.1.5 Drug Development Pipeline

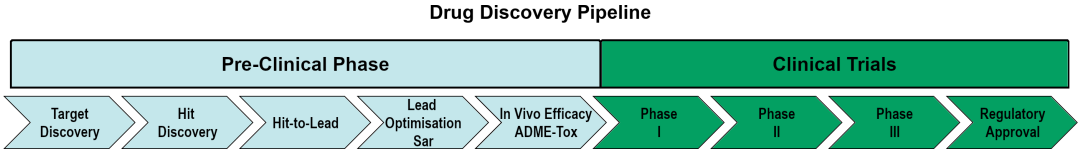


FIGURE 2.8: The two main Phases of the Drug Development Pipeline and their constituent stages.

The drug development pipeline (Fig. 2.8) is an interdisciplinary scientific process aiming to develop new pharmaceutical products with high levels of effectiveness and safety. This pipeline involves intricate steps for identifying, optimising, formulating and testing drug candidates to ensure they meet the rigorous standards required for market approval.

The modern drug development pipeline is composed of nine distinct stages that can be grouped into two main phases: the preclinical phase and the clinical phase. The preclinical phase is composed of five stages: Target Identification and validation, Hit Identification, Hit to Lead, Lead Optimisation / Structure-Activity Relationship, and In Vivo efficacy + ADME-TOX (absorption, distribution, metabolism, excretion, and toxicity). The first four stages can be loosely grouped into the discovery level, while the later stage can be considered to be part of the development level, in which the safety and efficacy of the new drug molecules are tested through homologous models before starting testing on the target organism. The Clinical Phase is initiated once pre-clinical evaluations confirm the drug's potential. This phase comprises four stages: phase I, phase II, phase III, and phase IV.

Target Identification and Validation

The process of target identification and validation begins by establishing a biological component, usually a protein, for the drug to target. This component should be a critical part of the disease pathway so that modulating it will disrupt the pathway and provide therapeutic benefits. This approach is known as target-driven drug discovery (TDD) [38].

Once proteins implicated in the disease pathway have been identified, researchers attempt to validate the targets to determine which protein(s) should be the focus of the drug discovery process. Validation seeks to confirm whether the biologically relevant protein is essential in the pathway, whether modulating the target will affect the phenotype, and whether the target is amenable to chemical modulation, known as druggability.

An alternative approach to TDD is the identification of targets through observations of their therapeutic effect on disease phenotypes; this is also known as phenotypic drug discovery (PDD) [307]. Historically, PDD has been the predominant method, constituting the methodology through which the majority of first-in-class drugs approved by the FDA between 1999 and 2008 were discovered [288].

Hit Identification

Hit identification is a crucial step in drug discovery, focusing on finding molecules that can modulate the target identified by the discovery project. Given the vast chemical space, with an estimated 10^{63} possible drug-like molecules [86], this task requires sophisticated methods and extensive resources.

One strategy for hit identification is searching public databases and patent literature for compounds known to bind to the target of interest or to a closely related target. This approach can leverage existing knowledge and expedite the initial stages of discovery, possibly leading to drug repurposing.

HTS is another of the key methods employed in hit identification. It involves the rapid testing of thousands to millions of compounds against the target and can efficiently generate a large number of hits.

Another one of the fundamental methods for hit identification is FBDD, in which fragments that weakly bind to the target are optimised into more potent drug candidates.

By integrating these strategies, researchers identify promising compounds for further development, forming the foundation for subsequent stages in the drug discovery pipeline.

Hit to Lead

The hit to lead stage involves the meticulous evaluation of hit compounds identified in the initial discovery phase. The goal is to develop a portfolio of lead compounds that exhibit improved characteristics and hold promise for further development.

During this stage, a multitude of compounds are assessed based on project-specific criteria, ensuring that they possess desirable attributes for future pharmaceutical applications. Key features evaluated include maintaining oral bioavailability, which is essential for drugs intended to be administered orally. The ability of compounds to cross the blood-brain barrier is another factor considered, especially for treatments targeting central nervous system disorders. Another factor considered is the estimated ease of synthesis in order to ensure that the compounds can be efficiently produced on a large scale. Other key features include toxicity and cell permeability. Compounds must demonstrate low toxicity to ensure safety and adequate cell permeability for effective drug delivery and action within target cells.

Alongside these objective features, a series of subjective metrics are employed to refine the selection process. One prominent metric is "drug-likeness", a loosely defined set of properties that compare potential lead compounds to existing drugs. This metric operates under the assumption that compounds more similar to already approved drugs have a higher likelihood of successful development. A well-known set of properties used to assess drug-likeness is the "Lipinski Rule of Five" [181], which provides guidelines on molecular properties that influence a compound's suitability as an oral drug.

By integrating these diverse criteria and metrics, researchers can compile a robust portfolio of lead compounds, each showing promise for future optimisation and development into safe, effective pharmaceuticals.

Lead Optimisation

Lead optimisation is the stage in drug development where lead molecules are modified to increase their potency and selectivity against the target, enhancing their pharmacological properties such as solubility, cost-effectiveness, ADME, and low toxicity [132]. This stage commonly employs iterative "design, make, test, analyse" (DMTA) cycles (Fig. 2.9), where each cycle involves designing modifications to the lead compound based on some hypothesis related to the link between chemical structure and bioactivity/molecular properties, synthesising the new variants, testing their pharmacological properties, and analysing the results in order update the hypothesis and inform the next round of design [14, 232]. This cycle repeats until a molecule that satisfies all the necessary bioactivity/selectivity/toxicity criteria is found.

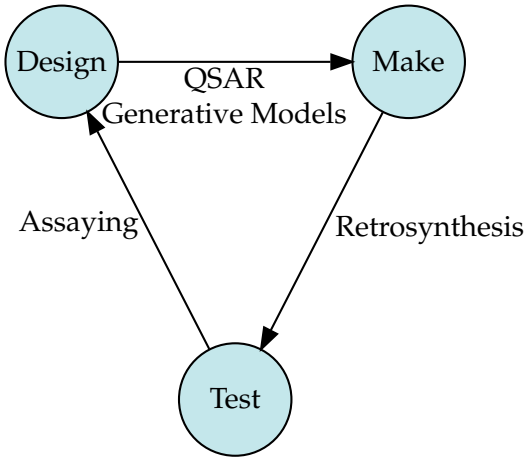


FIGURE 2.9: Lead optimisation (DMTA) cycle.

The design stage involves selecting and applying a predictive model to propose a set of new compounds likely to possess the desired properties. These models are often informed by quantitative structure-activity relationship (QSAR) studies, molecular docking, or other computational techniques that predict how alterations in molecular structure might affect bioactivity and pharmacokinetic properties.

Following the design phase, the make stage focuses on synthesising the proposed compounds; this stage relies heavily on a concept of synthetic organic chemistry named Retrosynthetic analysis. This concept was introduced by E.J. Corey [61] and involves deconstructing the target molecule into simpler precursor molecules through a series of logical steps until commercially available starting materials are identified. Corey's approach revolutionised the field, enabling the synthesis of complex molecules and earning him the Nobel Prize in Chemistry in 1990 [61, 292].

In the test stage, the biological activity of the synthesised compounds is experimentally verified using various biochemical assay techniques. These assays evaluate the efficacy, selectivity, and toxicity of the compounds, providing critical data that informs the next cycle of design.

The main paradigms for molecular optimisation in lead development are FBDD, ligand-based drug discovery (LBDD) and structure-based drug discovery (SBDD).

LBDD focuses on optimising molecular features by utilising data from other chemical structures and fundamental chemical principles without explicitly understanding the target's mechanism of action. By analysing the structure-activity relationships (SAR) of known compounds, LBDD aims to identify key molecular features that contribute to biological activity.

In contrast, SBDD utilises detailed three-dimensional structural information of protein-ligand interactions to inform the rational design of drug candidates. This approach focuses on the precise spatial configuration of both the target protein and the ligand to optimise binding interactions. Structural data for this can be obtained through experimental techniques such as X-ray crystallography or NMR spectroscopy. The structural information can also be obtained through computational methods and models such as AlphaFold 2. The high-resolution structural information allows researchers to identify potential binding sites and design modifications that improve ligand affinity and specificity. The advancements in high-throughput crystallography [26] have significantly enhanced the proactive application of SBDD in drug design.

Preclinical Testing

The preclinical testing phase aims to ensure a compound's safety and efficacy before it enters human trials. This stage seeks to establish whether the compound is safe for human administration and to determine initial clinical trial dosages. A primary focus of pre-clinical testing is characterising the pharmacokinetics (PK) of the compounds, particularly their ADME properties, which govern how the molecule traverses the body [155]. Understanding these properties is essential to predict the compound's behaviour in humans, ensuring proper dosage and minimising adverse effects.

Preclinical testing also involves running experiments to identify potential negative interactions with unsuitable biological systems. This includes evaluating the compound's effect on the hERG channel, which is critical for cardiac safety, and on CYP450 enzymes, which are essential for drug metabolism [132]. These tests help in identifying off-target effects that could lead to toxicity or adverse drug reactions.

In the later stages of preclinical testing, animal models, such as rats or mice, are used. Animal models are considered to be the gold standard for initial toxicity testing. These models are used to assess the compounds' toxicity profile and to extrapolate potential human response. However, it is important to note that while animal models provide valuable insights, they do not guarantee safety in humans [155].

Clinical Trials

Phase I is the first stage of the clinical trials. It typically spans several months and involves a small number of subjects, normally ranging from 20-100 volunteers [282]. With a success rate of around $60\% - 70\%$ [72, 282], this initial phase primarily aims to establish the baseline safety of the treatment in humans and to determine appropriate dosage levels for subsequent trial stages [282]. During these trials, healthy volunteers are administered the proposed treatment and closely monitored for adverse side effects. Notably, Phase I trials do not assess the treatment's efficacy in addressing the affliction in question.

Phase II of the clinical trials, lasting from several months to two years [282], involve significantly larger groups of volunteers than Phase I, typically encompassing several hundred patients who suffer from the disease targetted by the treatment [282]. The failure rate in Phase II is substantial, with approximately 65%-77% of compounds not progressing past this stage [72, 282]. Nearly 50% of these failures are attributed to a lack of efficacy [117], often due to incorrect hypotheses about the drug targets or the inability of the molecule to effectively modulate the target in vivo. The primary goal of Phase II trials is to determine the treatment's efficacy and to identify any side effects within a clinical setting. These trials are typically conducted double-blind; neither the patients nor the researchers know who receives the target drug or a placebo, ensuring unbiased results.

Phase III of the clinical trials typically spans 1 to 4 years and involves a significantly larger cohort of volunteers, ranging from 300 to 3000 participants [282]. With a failure rate of approximately 70% [282], this phase is crucial for assessing the effectiveness of the proposed treatment and meticulously monitoring adverse reactions [282]. These trials are essentially expanded versions of Phase II trials, designed to confirm the treatment's efficacy and safety on a broader scale. After the successful completion of Phase III, marketing authorisation can be sought, allowing the drug to be administered to patients outside of clinical trials [282].

Phase IV of the clinical trials, conducted once the drug has received approval, involves ongoing monitoring to assess the long-term safety and efficacy of the drug. These post-marketing studies aim to identify any side effects that may not have been apparent in earlier trial phases, evaluate the long-term risks and benefits, and confirm the drug's efficacy in a broader patient population. This phase ensures continuous surveillance and helps detect rare or late-emerging adverse effects, providing crucial data that can inform any necessary adjustments to the drug's use or dosing guidelines [282].

The rigorous and multifaceted nature of clinical trials highlights the complexity and meticulous planning required for successful drug development. Each phase, from initial safety assessments in Phase I to comprehensive evaluations in Phase IV, is critical in ensuring that new pharmaceutical products are both effective and safe for widespread use.

The drug development pipeline embodies a collaborative and interdisciplinary effort, integrating advances in biology, chemistry, pharmacology, and clinical sciences. Despite the high costs, extended timelines, and substantial risk of failure, the process is fundamental to bringing new therapeutic innovations to market, ultimately improving patient outcomes and advancing medical science.

2.1.6 Computer Aided Drug Design

Computer Aided Drug Design (CADD) denotes a suite of computational tools regularly used to study structure and function relationships and to search for new compounds at a relatively cheaper cost compared to using only experimental lab methods [329]. The name Computer Aided Drug Design indicates the use of computational techniques in order to enhance the drug design process. This process can be defined as the search for novel compounds with biological activity based on a systematic methodology [290]. CADD aims to predict whether a ligand binds to a protein and, if so, to determine the binding affinity of that ligand for the protein.

The potential of CADD lies in its ability to expedite the design of new drugs by replacing many expensive, time-consuming *in vitro* and *in vivo* experiments with efficient, cost-effective *in silico* ones. These computational methods scale better and allow larger regions of chemical space to be explored [225, 276].

The promise of CADD over experimentally-driven approaches can be demonstrated by the comparative screening for inhibitors of tyrosine phosphatase-1B [209]. In a virtual screen using docking, 365 compounds were suggested, with 127 achieving IC50 values of less than 100 µM, yielding a hit rate of nearly 35%. In contrast, high-throughput screening (HTS) of 400000 compounds tested suggested 81 compounds, resulting in a significantly lower hit rate of 0.021%. This stark difference illustrates the efficiency and effectiveness of CADD in drug discovery.

2.1.7 Virtual Screening

Virtual screening (VS) is a computational technique used to search through large ligand libraries to identify novel compounds with potential biological activity. This approach has become essential in modern drug discovery due to the financial and time constraints associated with experimental screening. Experimental screening processes, such as high-throughput screening, are notoriously expensive and time-consuming, often yielding low hit rates [163, 246, 279]. Virtual screening improves the efficiency and effectiveness of these experimental screens by refining the compound libraries to be tested, informing the optimisation of potential compounds [172, 279].

One of the primary advantages of virtual screening lies in its ability to rapidly and cost-effectively assess a vast number of compounds. This scalability allows researchers to explore larger regions of chemical space compared to traditional methods. By computationally evaluating compounds, virtual screening can prioritise

those most likely to exhibit the desired biological response, thereby reducing the number of candidates that need to be tested experimentally.

However, virtual screening is not without its limitations. It requires prior knowledge about the factors responsible for the desired biological response, which can sometimes be incomplete or inaccurate. While virtual screening can significantly narrow down potential candidates, its accuracy is not always sufficient to draw definitive conclusions.

There are two primary methodologies used in virtual screening: ligand-based virtual screening (LBVS) and structure-based virtual screening (SBVS).

Ligand Based Virtual Screening

Ligand-Based Virtual Screening (LBVS) employs a set of ligands with known binding properties to predict the binding affinity of new compounds. Through leveraging the features of these ligands, LBVS can infer whether a new compound will interact with a target, even in the absence of structural information about the protein target of knowledge of the binding mode.

LBVS can, in many ways, be considered an evolution of quantitative structure-activity relationship (QSAR) modelling, a concept pioneered by Cortwin Hansch in his studies of growth-regulating phenoxy acetic acids from 1962 to 1964. Hansch's work laid the foundation for QSAR by combining several physicochemical parameters into a single regression function, feining the lipophilicity parameter π, and formulating the parabolic model for nonlinear lipophilicity-activity relationship [160]. QSAR models use the physicochemical properties or theoretical descriptors of molecules to construct quantitative models predicting their biological activity. The ultimate goal of QSAR and LBVS is to build a scoring function that can accurately predict a molecule's properties or biological activity.

Several methodologies are employed in LBVS. Pharmacophoric models identify and match the spatial arrangement of features critical for binding [166]. Chemical similarity search methods find compounds with similar structures to known active compounds, based on the premise that similar molecules often exhibit similar biological activities [316]. Machine learning models, including convolutional neural networks, have become prominent in LBVS for their ability to learn complex patterns and make accurate predictions based on ligand features [317]. Convolutional Neural Networks, in particular, have shown success in predicting binding affinities and identifying potential inhibitors [124, 141, 244, 338].

A significant advantage of LBVS lies in its applicability without requiring structural information about the target protein, making it useful when the protein's structure is unknown or difficult to obtain. However, LBVS has notable limitations. The models are typically built for specific biological responses and are not easily generalisable to new targets. LBVS relies on interaction data for the target, necessitating some experimental groundwork. A generalisable model that can distinguish

between different proteins using only ligand-based features is theoretically unfeasible [32]. The high performance of ligand-based models on multi-protein databases suggests a strong ligand bias in the training and test sets, raising concerns about the models' reliability [32]. Additionally, the low interpretability of some LBVS models makes it harder to know if the model is truly learning the biophysics of protein-ligand interactions and molecular recognition or just reflecting ligand biases [32, 206].

Despite these challenges, LBVS remains a powerful tool in drug discovery, complementing structure-based methods and enhancing the ability to identify promising drug candidates efficiently.

Structure-Based Virtual Screening

Structure-Based Virtual Screening (SBVS) leverages detailed information about the protein target, typically its three-dimensional structure, to predict if a compound will bind to the protein [172]. This method is invaluable in the early stages of drug discovery as it can be utilised even in the absence of experimental binding data for the target of interest.

The SBVS process is typically multi-step, beginning with obtaining the 3D structure of the target protein [192]. This structure can be modelled from experimental results such as X-ray crystallography or NMR Spectroscopy. Alternatively, when experimental data is not available, the structure can be predicted from the amino acid sequence using computational methods. Establishing the protein's three-dimensional structure is crucial as it provides the framework for identifying potential binding sites for small molecules.

Once the 3D structure is obtained, the next step is to establish binding pockets. These binding pockets can be identified through experimental techniques or computational methods [339]. Accurate identification of binding pockets is essential as it sets the stage for the docking process, where potential ligands are evaluated for their binding affinity to the protein.

In the docking process, prospective ligands are docked into the identified binding pockets of the protein target. Multiple possible binding modes, or poses, of the ligand within the active site of the protein are proposed. These poses are then ranked to determine the most likely representation of a true binding mode, a process known as pose prediction. Traditionally, this ranking is achieved using an estimate of the binding affinity, but recent advancements have seen the application of machine learning techniques to train scoring functions for this task.

SBVS ultimately uses one or several potential poses for a protein-ligand complex to assess the likelihood that the compound will bind to the protein. By generating and evaluating these poses, SBVS facilitates the identification of promising drug candidates, thereby enhancing the efficiency and effectiveness of the drug discovery process.

2.2 Machine Learning, Deep Learning and Large Language Models

Machine learning is a branch of computer science that studies the use of algorithms that 'learn' patterns and decisions based on observed data without being explicitly programmed for specific tasks. The field integrates principles from statistics, data science and computer algorithms to create models that generalise from known to unseen data.

The conceptual roots of Machine Learning trace back to 1950 to pioneers such as Alan Turing, who speculated on the potential for machines to simulate human intelligence in his seminal paper, "Computing Machinery and Intelligence" [299], and Samuel A. L., who is widely believed to have defined the term of "Machine Learning" in 1959 as "a field of study that gives computers the capability to learn without being explicitly programmed" [264]. In his paper "Some Studies in Machine Learning Using the Game of Checkers" [264], Samuel A. L. introduces an early example of an adaptive algorithm that demonstrates the potential for programs to improve over time based on accumulated experience.

In 1997, Thomas Mitchell succinctly defined the essence of learning: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at a task in T, as measured by P, improves with experience E" [204]. This principle underscores the potential of Machine Learning across various domains.

Building on the historical foundations of machine learning, the field has grown into multiple branches depending on the types of data and learning objectives. Supervised learning stands out for using labelled datasets to train models that predict outcomes. In contrast, unsupervised learning focuses on patterns and relationships in unlabelled data. Weakly supervised learning represents a hybrid approach, utilising noisy or limited labels to train models where acquiring fully labelled datasets is challenging. While the mentioned techniques mainly focus on different types of identification, evolutionary algorithms mainly approach optimisation problems with a design inspired by nature. These diverse machine learning paradigms underpin the recent AI Revolution [118].

2.2.1 Genetic Algorithms

Genetic Algorithms (GAs) are a distinctive class of metaheuristic algorithms drawing inspiration from the Darwinian principle of natural selection. GAs belong to the broader family of Evolutionary Algorithms, which includes methods like Particle Swarm Optimization [233] and Ant Colony Optimization [79], each utilising unique biologically inspired metaphors, such as pheromones, reproduction and mutation, in order to solve optimisation and search problems. GAs function by imitating population-based evolution processes such as natural selection, reproduction and

mutation. These operations enable the algorithms' iterative evolution towards a solution.

One of the first researchers to experiment with machine learning was R. M. Friedberg, who, in his 1958 paper titled "A Learning Machine: Part I" [96], explored the process of modifying computer programs by introducing random changes and selecting those that enhanced performance on specific tasks in a manner reminiscent of modern GA approaches. Building on these early concepts, John Holland significantly advanced the field by formalising in his seminal 1975 work, "Adaptation in Natural and Artificial Systems" [125]. Holland's contributions have been essential in popularising the approach that is going to represent a cornerstone in the foundations of machine learning. [310]. Employing a nonlinear, discrete event, stochastic process in order to gain the capacity to tackle NP optimisation problems. Problems where either no polynomial-time solution is known or where a polynomial-time solution does not exist. This class of problems are inherently difficult due to the intractability of exhaustively exploring all possible solutions. In this context, GAs excel by balancing the opposing forces of exploration and exploitation. Exploration broadens the scope of the search to find a better-performing solution and to escape locally optimal solutions, while exploitation narrows the scope, focusing the search, in a greedy behaviour, on the vicinity of promising solutions. Through the adept handling of these competing priorities, GAs gain the capability to navigate complex solution landscapes.

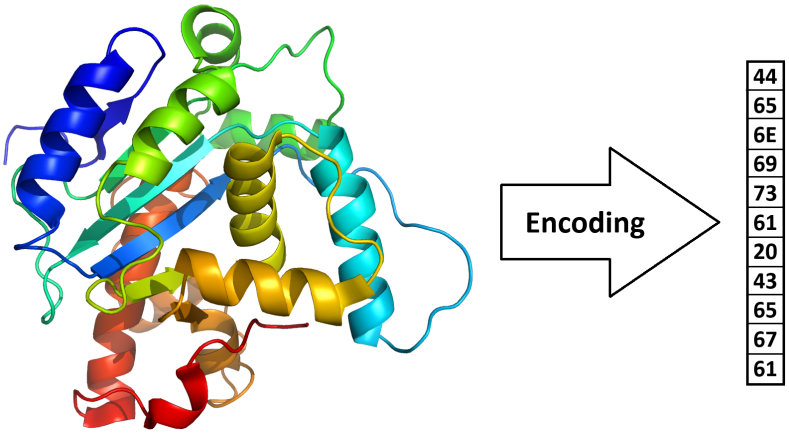


FIGURE 2.10: Illustration of the process of protein-encoding - the process of translating the real-world protein into numbers that can be processed by a computer.

One of the major choices in tailoring the GA to a specific task is represented by the choice of encoding mechanism (Fig. 2.10) or chromosomal representation of solutions. The choice of representation is crucial as it underpins the choice of genetic operators and drastically affects the performance of the GA. In order to maximise the performance of the search, the representational space should aim to be as restricted as possible in order to avoid spending computing power on the exploration

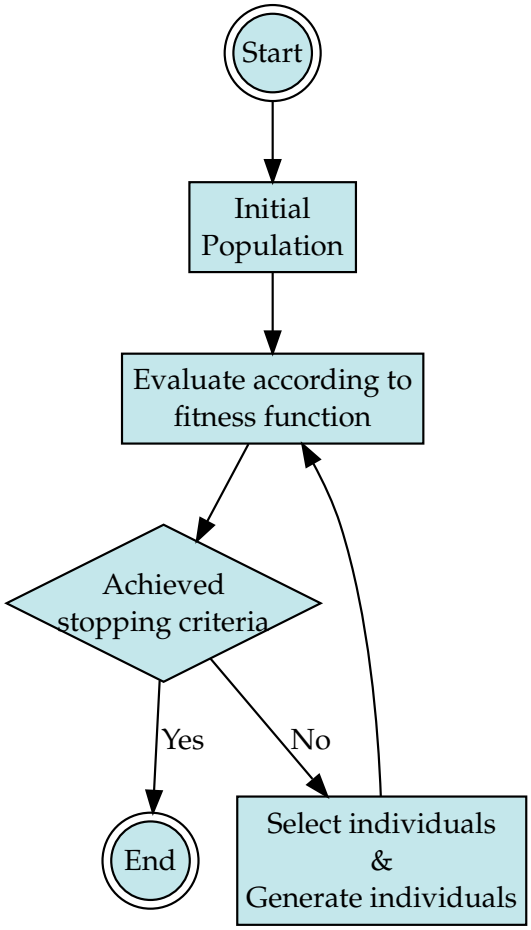


FIGURE 2.11: Flowchart depicting the general workflow for the classical implementation of GA.

of invalid solutions while being careful not to exclude any viable solutions or individuals that contain partial solutions. Another factor in the choice of chromosomal representation is constituted by the genetic operators that will be used in later stages; for example, the mutation operation benefits from a continuous encoding that conserves vicinities such that a small mutation leads to a small difference in the desired properties of the solution. The crossover operator benefits from an encoding scheme that allows individuals to be combined in a manner that enables properties from each of the parents to end up present in the offspring. The chromosomal representation of the solution, together with the design of the fitness function and the choice of genetic operators, are some of the main design questions in the tailoring of a GA to a concrete application.

GA Mechanism of function

The search process of GAs (illustrated in Fig. 2.11) works at a population level, iteratively transforming candidate solutions until a stopping criterion is met. In order to search for the solution, the GA starts from an initial population of possible solutions. These solutions are evaluated using a problem-specific fitness function that

evaluates the performance of each individual solution in the context of the given task. This evaluation guides the subsequent selection process, where individuals are selected based on their scores, favouring the top performers. Following selection, genetic operators are applied to the selected individuals to produce a new generation of individuals that replaces the previous generation. This cycle of evaluate, select and generate is iteratively repeated until a predefined stopping condition is reached, such as a target fitness value or a specified number of generations has been produced. The next paragraphs are going to go into more detail about each of the stages of the search process.

The initialisation of the initial population plays a crucial role in the efficiency and effectiveness of the search process. There is a diverse set of strategies for generating the initial solution candidates that form the initial population. One of the most common approaches is to create the population stochastically through the sampling of a uniform distribution across the entire search space. This randomness increases the diversity of the genetic pool and improves the exploratory powers of the GA. This sampling can be further enriched by prior knowledge of the solution space; the sampling distribution can be shaped such that promising areas receive more attention and areas. If partial or suboptimal results are available, these can be injected into the population in order to capture this prior knowledge. Another common strategy is to employ an intermediary or final population from a previous run to jumpstart the search process.

The next stage in the search process is constituted by the evaluation of the population of solutions to determine which individuals are top performers and possibly contain partial solutions; these individuals are going to be the main influencers in the creation of the next generation. This evaluation is conducted using a fitness function, defined as $f : i \to \mathbb{R}$, where each individual $i \in P$ of the population P is scored. The fitness function is a metric that quantifies the quality and effectiveness of the candidate solution represented by the individual. The design of the fitness function is inherently problem-dependent, requiring fine-tuning based on the specific objectives and constraints of the application. The fitness function can take a multitude of different forms, ranging from numerical evaluations to heuristic approximations and even computational simulations. In scenarios requiring intense computational resources, the scoring stage can be executed in a distributed environment.

Once the evaluation of the current population has been effectuated, the GA verifies if any of the stopping criteria have been achieved. If the termination conditions have been fulfilled, the algorithm usually returns either the individual that matches the stopping requirements or the top-performing individual. Some of the most common stopping criteria are achieving or surpassing a preset fitness function value or reaching imposed limits for the number of generations or processing time. If none of the stopping conditions have been met, the algorithm proceeds to the selection stage.

In the selection stage, individuals are chosen from the current population to contribute to the next generation. This selection is typically stochastic in nature, biased towards individuals with higher fitness scores, employing a Darwinian strategy of "survival of the fittest" in order to promote the perpetuation of advantageous traits. However, it is crucial to maintain genetic diversity within the population to ensure a thorough exploration of the solution space. Therefore, lower-performing individuals are not entirely excluded from the selection process as these individuals might contain unique genes or partial solutions not found in their higher-scoring counterparts. To safeguard the achievements of the best-performing individual from the threats of the stochastic nature of the selection and from changes inflicted by the genetic operators, the strategy of elitism is often employed. This approach ensures the preservation of the top performer through direct promotion to the next generation, ensuring that the genetic integrity of the best solution is maintained. The strategy of elitism often enhances the stability of the search progression.

The generation of an offspring population constitutes the next and last stage of the GA search cycle. Through the repeated application of genetic operators on selected individuals, new individuals are generated containing both inherited and novel genetic combinations. These operations are essential for introducing variation and, together with the pressures exercised by the selection, shifting the population towards better-performing individuals.

Genetic Operators

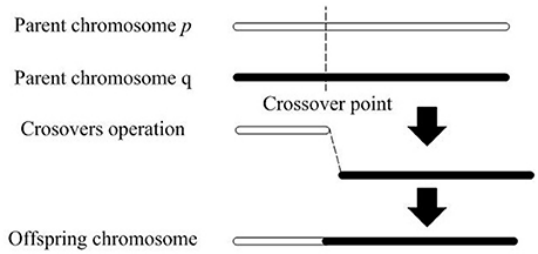


FIGURE 2.12: Illustration of the single-point crossover genetic-operator.

One of the fundamental genetic operators is the crossover operator (Fig. 2.12), also known as the recombination operator. This operator combines genetic information from two or more [83, 295] selected 'parent' individuals to produce new 'offspring'. This process is strongly tied to the chosen genetic representation of the solutions. For instance, the one-point crossover operation involves two parent solutions $p_a, p_b \in P$, encoded by the gene sequences $p_a = (a_1, a_2, \ldots, a_m)$ and $p_b = (b_1, b_2, \ldots, b_m)$. A crossover point n is randomly chosen where $0 \leq n \leq m$ in order to generate usually two new offspring that combine traits from both parents $p_a \circ p_b = (a_1, a_2, \ldots, a_n, b_{n+1}, b_{n+2}, \ldots, b_m), (b_1, b_2, \ldots, b_n, a_{n+1}, a_{n+2}, \ldots, a_m)$. There are a multitude of recombination strategies, such as one-point crossover, K-Point crossover, Uniform Crossover, Average Crossover etc., as illustrated by Umbarkar

et al. [301] that aim to preserve advantageous traits or partial solutions from both parents and potentially create a full solution or at least a more fit individual.

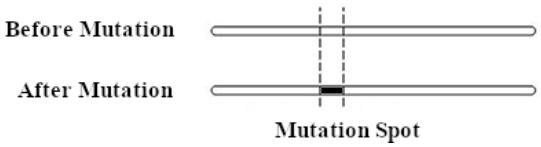


FIGURE 2.13: Illustration of the single-point mutation genetic-operator.

The mutation operator (Fig. 2.13) is another of the fundamental genetic operators commonly used in GA applications. This operator is pivotal for the exploration of the solution space as it introduces new genetic material into the population by altering existing genes and thus providing a mechanism to explore new areas of the solution space that have not been covered by recombination alone. This diversity is essential for combating the homogeneity induced by selection pressures and preventing the algorithm from prematurely converging on suboptimal solutions. One of the most simple mutation techniques often used in applications is point mutation. This technique involves a predetermined mutation value m known as mutation rate. Given one individual i with its genetic representation $i = (a_1, a_2, \ldots, a_m) \in P$, m spots are randomly selected to undergo mutation. This mutation can result in a gene being flipped or changed, leading to a new variant of the individual $i' = (a_1, a_2, \ldots, a'_i, \ldots, a_m)$. Point mutation is just one of the many mutation strategies, such as directed mutation, mirror mutation, covariance matrix-based mutation, etc. [280]. All these strategies aim to increase the diversity of the population and to enhance the algorithm's capability to thoroughly explore the solution space through the injection of new genetic material.

GAs are subject to several limitations that can severely affect their efficacy. One significant challenge is presented by the computational intensity associated with repeated evaluations of the fitness function, especially in cases where the function involves complex simulations or multi-dimensional data analysis, which can make the process resource-intensive and time-consuming. Another condition imposed on the landscape generated by the fitness function. Landscapes with little or no gradient, such as the ones generated by binary functions, offer too little information for the evolutionary process to be viable, severely stalling the optimisation process. Another of the complexities of the GA is raised by the choice of the strategy for genetic encoding. The encoding scheme plays a crucial role in the efficacy of the GA, as using a too-broad scheme can waste computing resources on invalid solutions, while a too-restrictive encoding can exclude potential solutions or partial solutions that would contribute to the finding of an optimal solution. The ideal encoding scheme captures the essential complexities of the problem space and focuses the search only on viable options while offering enough flexibility for partial solutions.

Despite their limitations, GAs have been used in diverse applications across various domains, demonstrating their versatility and efficacy. Wang et al. [311] emphasise the extensive use of GAs in fields like Resource Management and composite material optimisation, showcasing their effectiveness in optimising complex processes and configurations. Similarly, Katoch et al. [146] highlight the broad utilisation of GAs in areas such as Scheduling, Facility Layout and Gaming. These examples illustrate the GAs' capacity to handle a wide range of optimisation and search problems, making them invaluable tools.

2.2.2 Supervised Learning

The essence of the supervised learning approach lies in teaching computers to predict outcomes based on a labelled dataset $S = ((x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)) \in X \times Y$, where x_i is usually a multidimensional vector corresponding to a data point and y_i, represents the target value or class.

Most common techniques for achieving supervised learning in modern times consist of learning the parameters Θ of a predictive model $y = f(x, \Theta)$ that minimises a given loss function $L(y, \hat{y})$, where x is the input and can range from a vector of values to images, videos, text or even raw sensor signal data. y is the target class (in case of classification), or the target value (in case of regression), and $\hat{y}$ is the class or value predicted by the model.

The loss function $L : Y \times Y \rightarrow \mathbb{R}^{\geq 0}$ is one of the components that plays a crucial role in any supervised learning algorithm. It has the role of telling quantitatively how to correct the model is by measuring the similarity between the $f(x, \Theta)$ prediction and the y target value. Depending on the nature of the target value y, two major distinct classes of objectives emerge with different loss functions. The case that Y is continuous denotes a regression scenario where the target space Y can be considered to be a multidimensional real number, $Y = \mathbb{R}^m$. Table 2.1 illustrates some of the most common regression loss functions and their formulas. The choice of loss function is highly dependent on the problem to be solved, the distribution of the target variable and the tolerance for outliers. For example, MSE can be more suitable when it is critical to penalise larger errors heavily but not have to deal with many outliers, while MAE may be preferred when a linear penalty for errors is desired with higher robustness to outliers. The case that Y is discrete indicates a classification scenario with $|Y|$ indicating the number of distinct classes. For this objective, some of the most common loss functions can be found in table 2.2. A more detailed analysis of the loss functions can be found in Qi Wang et al. paper "A Comprehensive Survey of Loss Functions in Machine Learning" [310].

The process of 'learning' in supervised learning, also known as training, is fundamentally an attempt to identify an optimal set of parameters, denoted as $\hat{\Theta}$, that minimises the loss function L. This optimisation is formalised as finding the solution to $\hat{\Theta} = arg \min_{\Theta} L(y, f(x, \Theta))$, where the goal is to find the parameter values that result in the least discrepancy between predicted outcomes and actual targets. To achieve

Common Regressions Losses	
Loss Function	Loss Formula
Mean Squared Error (MSE) / L2 Loss • Penalises larger errors more heavily due to squaring • Sensitive to outliers	$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$
Mean Absolute Error (MAE)/L1 Loss • Linear penalty for errors • Less sensitive to outliers	$\frac{1}{n}\sum_{i=1}^{N}\lvert y_i - \hat{y}_i\rvert$
Root Mean Squared Error (RMSE) • Similar to MSE; provides error in the same units as the target variable • Sensitive to large errors	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$
Huber Loss • Combines MSE and MAE • Requires choosing a threshold δ	$L_\delta = \begin{cases} \frac{1}{2}(y-\hat{y})^2 & if\ \lvert(y-\hat{y})\rvert < \delta \\ \delta((y-\hat{y}) - \frac{1}{2}\delta) & otherwise \end{cases}$

TABLE 2.1: Table defining some of the most common regression loss functions. y_i is the target value and $\hat{y}_i$ is the predicted value

Common Classification Losses	
Loss Function	Loss Formula
Binary Cross-Entropy Loss • Used for binary classification tasks • Penalises the distance between predicted probabilities and actually class labels	$-\frac{1}{N}\sum_{i=1}^{N}[y_i log(\hat{y}_i) + (1-y_i) log(1-\hat{y}_i)]$
Categorical Cross-Entropy Loss • Used for multi-class classification tasks • Similar to binary cross-entropy but for multiple classes	$-\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{M} y_{ic} log(\hat{y}_{ic})$
Softmax • Converts logits to probabilities for multi-class classification • Ensures that the output values are in the range $(0, 1)$ and sum up to 1	$\frac{e^{\hat{y}_i}}{\sum_j e^{\hat{y}_j}}$
Hinge Loss • Used for binary classification, especially in the case of Support Vector Machines • Encourages a margin of separation between classes δ	$\frac{1}{N}\sum_{i=1}^{N} max(0, 1 - y_i \cdot \hat{y}_i)$

TABLE 2.2: Table defining some of the most common classification loss functions. y_i is the target value and $\hat{y}_i$ is the predicted value

this, a variety of techniques have been developed, such as: recursive partitioning used by decision trees to simplify decision spaces, making them robust and interpretable albeit prone to overfitting; the optimisation of decision boundaries through quadratic programming used by Support Vector Machines, that makes them excel for linearly separable objectives; or the gradient-based optimisations used by deep neural networks that makes them top performers in modelling complex functions but compromises on the explainability, causing the black-box nature characteristic for deep neural networks.

The defining component of supervised learning is the labelled dataset S. The efficacy of supervised learning hinges on the quality and characteristics of the dataset. It not only provides the foundational input for model training but also shapes the learning algorithm's ability to generalise to unseen data. Datasets must be carefully curated to represent the problem space accurately, with sufficient variability and balanced classes in order to train robust models; the presence of noisy, incomplete, or imbalanced data can dramatically affect the learning process, requiring strategies such as data augmentation, imputation and balancing techniques to mitigate their effects. Furthermore, the size of the dataset is the characteristic that heavily influences the choice of the model; larger datasets pertain to more complex models without the fear of overfitting, whereas smaller datasets might require simpler models or techniques like transfer learning [12, 133]. The interaction between dataset characteristics and model selection is a critical factor in supervised learning.

Shallow Learning

Shallow learning, also known as traditional machine learning or classical machine learning, represents a foundational approach within the broader domain of machine learning by utilising relatively simple but powerful mathematical models to learn. These approaches often rely on handcrafted features extracted from data. The process of extracting meaningful features from existing datasets involves identifying and engineering features based on domain knowledge, intuition, and empirical experimentation. These features are of utmost importance as they directly influence the effectiveness of the model to interpret the underlying structure of the dataset. Depending on the nature of the features and the target objective, some of the foundational solutions in the field of classical machine learning are Support Vector Machines, Decision Trees/Random Forests and the Perceptron.

Support Vector Machines

Support Vector Machines (SVMs) are a type of classical machine learning algorithm introduced by Corinna Cortes et al. in their 1995 "Support-Vector Networks" publication [62]. Initially developed for binary classification tasks, the essence of SVMs lies in their unique approach to identifying a hyperplane in a high-dimensional

space that separates data points into two distinct classes while maximising the margin. The maximisation of the margin is one of the factors that confers SVMs the ability to generalise well on unseen data. In their original form, due to their approach, SVMs were only capable of handling linearly separable classes, greatly reducing the real-world application of the models. This limitation was addressed by the clever use of a technique introduced by M. Alizerman et al. in 1964 [5] known as the kernel trick, a method in which the features space is transformed into a higher dimensional space that facilitates the linear separability of the classes. The use of dedicated kernel functions [135, 136, 174], such as the polynomial or radial basis functions, extends the ability of the SVMs to model real-world, non-linearly separable datasets without computing the high-dimensional transformations. The success and diversity of the kernel functions highlight a notable flaw in SVMs: their sensitivity to the choice of their hyperparameters, especially in cases where the optimal parameters are not known beforehand. The search for optimal hyperparameter settings is often computationally intractable, especially for large datasets.

The SVM mechanism of functionality aims to solve for the weights w and bias b such that the hyperplane defined by $w \cdot x + b = 0$ maximises the margin between the classes. This is formalised as minimising $\frac{1}{2}||w||^2$ subject to the constraint $y_i(w \cdot x_i + b) \geq 1, \forall i$ where y_i are the class labels. For non-linear classification, SVMs employ kernel functions, $K(x_i, x_j)$ to project the input features into a higher-dimensional space without explicitly computing the transformation, thereby facilitating linear separation in the new space. Some of the most common kernels include the linear $x_i \cdot x_j$, polynomial $(\gamma x_i \cdot x_j + r)^d$, and radial basis function (RBF) $exp(-\gamma||x_i - x_j||^2), \gamma \geq 0$ kernels, each allowing the SVM to adapt to different data distributions.

Despite their limitations, SVMs have historically seen great success in the field of bioinformatics. Their ability to handle high-dimensional datasets coupled with the versatility of the kernel functions has made them particularly effective in classification tasks such as identifying protein sequence's remote homologies [135, 136], protein fold recognition [73, 174], cancer tissue classification [112, 302]. SVMs have also been used in prediction tasks such as protein-protein interaction [27] and secondary structure prediction in proteins [129]. These successes highlight the representational power of SVMs and their historical impact, establishing them as a foundational block in the field of classical machine learning.

Decision Trees And Random Forests

Within the domain of classical machine learning, Decision Trees stand as quintessential models, offering both precision and interoperability. These models excel in parsing the dataset through a series of binary decisions, leading to the iterative branching that gives this particular solution its namesake. At each node within this hierarchical structure, a decision is made to split the data based on a feature that best segregates the dataset into homogeneous subsets guided by different metrics. Two of the most popular metrics are Information Gain (eq. 2.1 where p_i is the probability of randomly

selecting an example in class i), a metric which quantifies the reduction in entropy achieved by a dataset split popularised by Quinlan J. R. [241] and founded on the Entropy theory of Shannon C. E. [275]; and the Gini impurity (eq. 2.2), introduced by Corrado Gini [45] which is used as a measure of impurity or diversity. Despite their advantages, Decision Trees are prone to overfitting. This condition causes the model to become too complex and attempt to capture noise in the dataset rather than the underlying patterns, leading to poor generalisation on unseen data.

$$\begin{aligned} Information\ Gain &= Entropy_{parent} - Entropy_{children} \\ Entropy &= -\sum_{i=1}^{n} p_i \times log_2(p_i) \end{aligned} \tag{2.1}$$

$$G = \sum_{i=1}^{C} p(i) * (1 - p(i)) \tag{2.2}$$

To mitigate the overfitting tendencies inherent in Decision Trees and strengthen their robustness, Leo Breiman [35] introduced and popularised an ensemble technique that builds upon the work of Tin Kam Ho [122] named Random Forests. Ensemble learning is a method that combines multiple models to improve overall performance. In order to achieve this, Random Forests generates a multitude of Decision Trees, each trained on a bootstrap sample of the data and selecting a subset of features. The final result is obtained by aggregating the responses from each Decision Tree through majority voting for classification tasks or averaging for regression tasks. By using a large number of trees trained on different subsets of the data, Random Forests can achieve a lower variance and, hence, improved performance over the utilisation of a single Decision Tree. They are relatively easy to use, requiring little extra hyperparameters compared to a single Decision Tree.

While Random Forests gain in robustness and predictive accuracy over a single Decision Tree, this improvement comes at the cost of lower explainability. The aggregated decision-making process, though effective, makes it more challenging to understand the specific logic behind each prediction. Furthermore, Random Forests inherit one of the fundamental limitations of Decision Trees: they struggle to extrapolate predictions beyond the range of the training data. This is because both methodologies rely on the partitioning of the existing data space without assuming any underlying functional pattern of the data.

In bioinformatics, Random Forests have emerged as one of the proven statistical models, enabling breakthroughs across diverse applications. Their precision in epitope prediction classification [335] showcases their ability to identify critical antigenic determinants accurately; their usage in identifying gene-gene interactions [183] and gene expression classification [87] illustrates their capability of dealing with complex features and interactions. Random Forests have also been used for proteomic classification [103], drug resistance prediction based on DNA sequences [64],

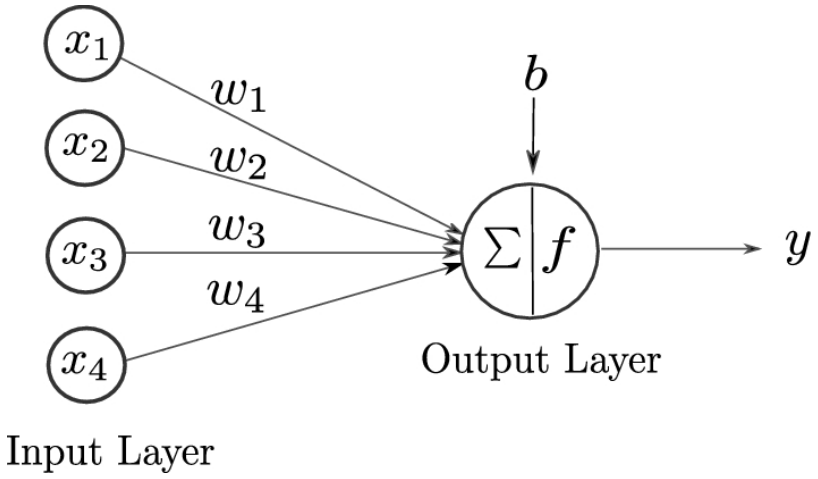


FIGURE 2.14: Schematic of the functionality of a single perceptron with four inputs, $x_1, x_2, \ldots, x_4$, with their afferent weights $w_1, w_2, \ldots, w_4$, the bias b. Each input is multiplied by its corresponding weight, the bias is added, and the result is passed through an activation function in order to get the output y.

and QSAR modelling [234, 287]. This plethora of applications demonstrates the versatility and effectiveness of Random Forests.

Perceptron

In 1957, Frank Rosenblatt proposed the model that is going to underpin most of the current breakthroughs in Artificial intelligence, the Perceptron [253], the building block that would become the cornerstone of neural networks and, by extension, modern AI. Inspired by the believed functionality of a neuron, Rosenblatt's model embodied an early attempt to simulate the thought process of the human brain. His work constituted early evidence of the machine's capability to learn sophisticated patterns and make decisions based on data inputs, marking a historical milestone in the development of modern Artificial Intelligence.

The basic mathematical model of the perceptron (fig. 2.14) works by performing non-linear transformations of its inputs. The model takes as inputs n descriptors or features $x_1, x_2, \ldots, x_n$ and outputs a prediction y following the $\sigma((\sum_i w_i \cdot x_i) + b) = y$, where σ is the activation function, w_i represents the weight for each feature input x_i, and b is the bias. The weights have the role of modelling the response of the perceptron to different input features. The weights and bias collectively form the Θ, trainable parameters of the model. These are the parameters that are adjusted during the training session in order to obtain a model with predictive capabilities. These parameters are initialised with random values from a normal distribution and iteratively updated according to the rule $W_{t+1} = W_t + \Delta_{W_t}$ with $\Delta_{w_i} = \eta(y - \hat{y})x_i$ where w_i is the weight associated with the input x_i, η is the learning rate, a small positive constant hyperparameter that determines the magnitude of the update, y is the true label, $\hat{y}$ is the predicted label.

One of the earliest applications of the perceptron in molecular biology was proposed by Stormo et al. [284] using the perceptron algorithm to locate the initial translation sites in E. Coli. This pioneering effort provided a computational method for a biological problem and underscored the potential of machine learning in genomics. Despite this early demonstration, the perceptron suffers from the same limitations as the basic form of SVMs. It is limited in modelling relatively simple problems with linearly separable solutions; thus, in the protein landscape, where patterns are often complex and nonlinear, the utility of the Perceptron is limited.

2.2.3 Deep Neural Networks

Deep learning is a subset of machine learning that revolves around the utilisation of artificial neural networks with the aim of facilitating direct learning from data. This approach differs from shallow learning methods, which traditionally require hand-crafted features refined by domain experts. In contrast, deep learning techniques are designed to discover and identify intricate patterns and correlations in large datasets automatically by leveraging multi-layered architectures that offer the models complex hierarchical representational power. This self-discovery and progressive abstraction capability allow end-to-end learning, where raw data can serve as the input, and the system can handle feature extraction, transformation, and predictive tasks. As a result, deep learning has emerged as a foundational pillar of machine learning.

Deep Learning has proven exceptionally effective in fields with complex data, leading to unprecedented advancements across a variety of applications. In computer vision, deep learning has succeeded in surpassing human-level performances in tasks such as image classification and detection [76] [120]. Some of the notable models include ResNet [119], ImageNet [157], and Clip [245]. Similarly, in natural language processing, sequence-to-sequence learning frameworks [119] and later transformative models like GPT-3 [39] have pushed the boundary of machine understanding and text generation. Speech recognition models such as Deep Speech [54] and WaveNet [269] have significantly improved the interpretation accuracy of human speech. Furthermore, in the domain of bioinformatics, deep learning has enabled unprecedented advances [265]. One of the most notable achievements in this domain was the solution of a challenge that has stood for half a century, the problem of protein structure prediction that AlphaFold 2 [325] solved. Each of these advancements clearly demonstrates the power and versatility of deep learning techniques in dealing with complex data across disciplines.

The transformative power of deep learning architectures stems from their underlying architectural design, which allows the hierarchical composition of simpler, modular building blocks. This layered approach enables the network to extract progressively complex patterns from the data, building from basic features to increasingly abstract and complex patterns with depth. This fact can be most evidently

observed in Convolutional Neural Networks, where initial layers are primarily concerned with the detection of edges and textures. In contrast, deeper layers focus on ensembles of textures and edges forming structures and substructures, as illustrated in Fig. 2.19. This depth is what separates it from Shallow Learning and is the central characteristic that lends its name to this category of machine learning. This depth gives deep learning models the ability to autonomously generate features without requiring human intervention [108], and even allows the models to tap into previously inaccessible sources of information, unstructured data [4].

Through the strategic composition of relatively simple building blocks, deep learning approaches function as general function approximators [127], supporting the learning of complex correlations between the input features and the targetted output. This flexibility gave birth to a plethora of different neural network designs, such as Convolutional Neural Networks (CNNs), which integrate convolutional operations in order to take advantage of its translational equivariance for computer vision tasks, two representative applications are AlexNet [157] and ResNet [119]; Recurrent Neural Networks (RNN), designed to learn temporal dependencies and utilised in text [54] and speech [182]; Long Short-Term Memory (LSTM) [123], enhances sequence learning and memory capabilities through the improved capturing of long-term dependencies; Gated Recurrent Units (GRU) [58], attempts to address the vanishing gradient issues; and Geometric Deep Learning, extends deep learning techniques to graph and manifold structured data. This vast modelling capacity that underpins the vast success of deep learning models can only be achieved in the presence of sufficient amounts of data.

One of the main facilitators of the current revolution within the field of AI is the introduction and availability of large-scale, domain-specific datasets. These data repositories and their quality have been essential in training performant models capable of learning complex patterns and tasks. Some of the more historically influential datasets are the Modified National Institute of Standards and Technology (MNIST) database [167], one of the most widely known datasets in machine learning containing 70,000 examples of handwritten digits; CIFAR-10 and CIFAR-100 datasets containing 60,000 colour images split in 10 respectively 100 classes; COCO [178], a large-scale object segmentation dataset consisting of 328,000 images; ImageNet [157], a database designed for visual object recognition with more than 14 million hand-annotated images belonging to over 20,000 categories. With the increase in model size, traditional, manually annotated datasets started to contain insufficient data for training and novel avenues for acquiring data were sought, such as Social Media posts and IMDb Reviews culminating in the Foundational Models that are trained on internet-level amounts of data. The growth in both scale and sophistication of these datasets from MNIST to internet-scale corpora tightly follows the advancements in machine learning models. The datasets form a foundational pillar for the success of deep learning models.

Another of the foundational pillars is engineering, which plays a crucial role

in the evolution of the models. The rapid advancements in computational power driven by the development of specialised hardware such as Tensor Processing Units (TPUs) together with the repurposing of existing Graphical Processing Units (GPUs) hardware specialised in massively parallel processing have significantly reduced the time required for training and inference, enabling not only the acceleration of the pace of innovation through more rapid prototyping but also the training of bigger and more complex models. The engineering progress has been indispensable for the training of deep learning models. It is also playing a pivotal role in the processing and storage of the massive datasets required for modern deep-learning solutions.

Despite their remarkable advancements across various domains, this approach harbours inherent limitations, limitations that are even more evident in the field of bioinformatics. One of these limitations is widely known under the name of Out-Of-Distribution prediction [179]; it refers to the struggle of the model to generalise on data that is substantially different from the training set. This generalisation gap presents a significant hurdle that is magnified in the field of bioinformatics due to the immense and largely unexplored space of protein structures. These unexplored spaces contain potentially groundbreaking drugs that are momentarily beyond the reach of current models due to their generalisation issues.

Another significant limitation of Deep Neural Networks lies in their lack of interpretability, also referred to as their "black box" nature. The ability of this type of model to mix and generate features from their input, coupled with their tendency to improve in performance with more input data and increased depth, makes unravelling the decision-making process very opaque. The inability to discern the logic behind the network's choices leaves the model open to the "clever Hans effect" [164], producing the correct outcomes based on faulty reasoning. The unexplainabilty of these types of solutions represents a significant hurdle in their application in clinical settings, where the model has to be accurate, consistent and have the correct reasoning for a diagnostic. The work of Jianping Zhang et al. [333] and Andrei Chertkov et al. [51] illustrate this issue; through specially crafted adversarial inputs, they demonstrate the unpredictability of modern models. The nascent field of AI Explainability addresses this limitation through the development of tools and methodologies that pierce the "black box" nature of deep neural networks, striving to make their statistical reasoning accessible and transparent.

2.2.4 Multi Layered Perceptrons

Multi-layered perceptron (MLP), also known as a Feedforward network (FFN, Fig. 2.15) represents a foundational architecture in the field of Deep Learning. Characterised by its composition of sequentially interconnected perceptrons or neurons, these networks are typically organised into distinct layers. Each perceptron within a layer receives inputs from the predecessor layer l_{i-1}, and forwards its output to its successor layer l_{i+1}. This hierarchical layering empowers the model to extract and transform features at increasing levels of abstraction as data progress through the

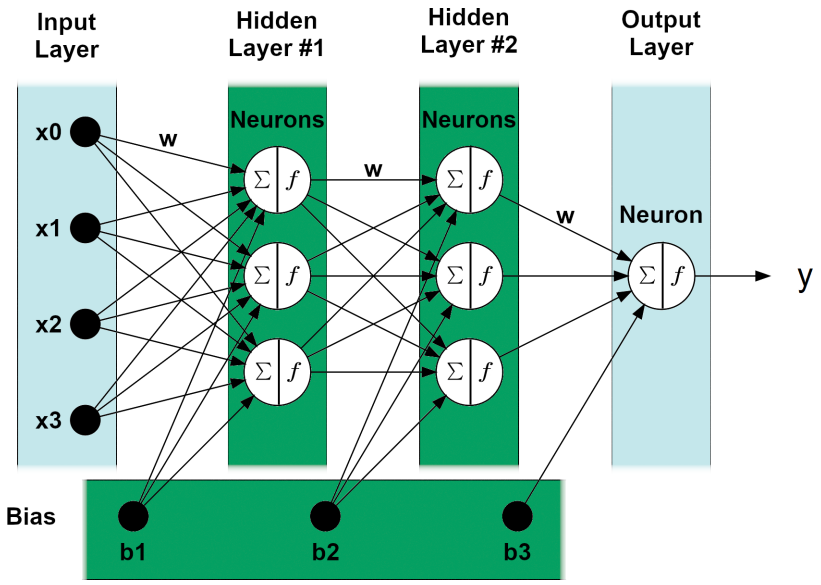


FIGURE 2.15: Layered illustration of the layout and connections of a 2 hidden layer MLP with 4 inputs, 3 perceptrons per hidden layer and one output.

network, boosting their representational power and allowing them to be universal function approximations [127].

The remarkable capacity of the MLP to gain representational power through iterative composition is enabled by the activation function σ. The activation functions introduce the essential non-linearity that allows the transcendence of linear mapping through hierarchical composition. Their necessity is essential and proven by the property of closure under composition for linear transformation. Without the activation function, the representational power of the model would not increase with depth, as the composition of two linear transformations is itself linear. Some of the most common activation functions include the Sigmoid activation function [256], the original function for the Multi-layered Perceptron, which maps any real-value number to the $(0, 1)$ interval; the Hyperbolic Tangent (tanh) Function [223], which compresses any real value to the $(-1, 1)$ interval; Rectified Linear Unit (ReLU) [3], one of the most widely used activation functions in modern neural networks due to its computational efficiency and ability to diminish the vanishing gradient problem.

Training a Multi-layer Perceptron, as with any other deep network model, revolves around the concept of adjusting its parameters to minimise a similarity function that measures the correctness of the prediction. This update is usually realised by employing techniques based on gradient descent over the topology defined by its loss function. The usage of gradients dictates one foundational limitation of Deep Learning approaches: the necessity of all mathematical operations within the network to be differentiable, ensuring that the contribution of each perceptron to the result can be computed and the unit can be updated according to its contribution. This highlights a significant limitation of Deep Learning approaches: the constraint of a differentiable loss function.

The reliance on gradient-based training methods introduces additional limitations. While deep networks can minimise the error on the training data, there is no guarantee that the solution found is optimal [106]; due to the nature of this type of training, for complex landscapes, the model is predisposed to get stuck in local minima. Another hurdle that severely limits the depth of these models, and therefore their representational power, is the issue of the gradient vanishing or exploding [20]. In these scenarios, the gradients can become excessively small or large as the corrections are transmitted back through the network, leading to ineffective or unstable training. These challenges highlight the intricate process of designing performant neural network architectures.

2.2.5 Gradient Descent

Gradient Descent stands at the core of deep learning and plays a fundamental role in updating the network parameters. It is the active mechanism through which the process of "learning" is achieved. This methodology can be classified as a first-order, iterative optimisation algorithm to find a local minimum for a differentiable function. Its foundational introduction by David E. Rumelhart et al. in 1986 [256] marked a cornerstone moment in the history of deep learning. By methodically tuning the weights, this method empowers neural networks to iteratively reduce the error dictated by the loss function, embodying the very essence of "learning".

The process of updating the weights of a neural network with the help of gradient descent starts by selecting a random initial point for the parameters denoted as $\Theta_0 \in \mathbb{R}^m$ where m represents the number of tunable parameters in the network. The initialisation of Θ_0 is critical; it requires randomness in order to break any potential symmetries between perceptrons. Without the breaking of these symmetries, every unit would behave identically, and the network would not be able to consider different aspects of the input features. The initialisation of Θ_0 must be chosen from an appropriate distribution of values in order to minimise potential gradient issues during the training [116, 120].

Following the initialisation phase, the learning process unfolds iteratively. At each iteration step t, the algorithm advances by taking a step, parametrised by the learning rate η, in the opposite direction to the gradient of the loss function L, as described by the update rule $\Theta_{t+1} = \Theta_t - \eta \nabla_\Theta L$. This methodical update of the parameters is usually repeated, reducing the error as defined by the loss function L, until the value of L has converged to a satisfactory number.

A key aspect of this iterative process is the stochastic approximation of the gradient $\nabla_\Theta L$. Instead of computing the true gradient across the entire training set, which would be computationally intensive, it is common practice to use an approximation of the true gradient obtained through computing the gradient on a random subset of the training set; this approximation is called a stochastic gradient. This approximation substantially improves the efficiency of the optimisation process by increasing the frequency of the updates, which is essential, especially in the first

stages of the training when the network is forming its rough approximation of the target function. This stochastic process also introduces variability that can help the model escape local minima in a process similar to simulated annealing [305]. Additionally, employing a stochastic gradient makes it faster to train on larger datasets by minimising memory access during the computation of the gradient. In such cases, one training iteration through the whole dataset is called an epoch.

At the heart of Gradient Descent lies the backpropagation algorithm, which effectively applies the chain rule to compute the gradient of the loss function with respect to each parameter of the model. The chain rule allows the derivative of a composite function to be expressed in terms of the derivatives of its constituent functions. Given a function $y = f(g(x))$, the chain rule states that $\frac{dy}{dx} = \frac{dy}{dg} \cdot \frac{dg}{dx}$, a principle heavily used in the process of backpropagation. This allows the efficient computation of the gradient of the loss function $\nabla_\Theta L$ for each layer, starting from the output and propagating back to the input, systematically computing the contribution of each parameter to the overall error. The propagation from outputs towards input gives the name of the method, backpropagation.

Most modern computational frameworks implement automatic differentiation; this key approach leverages computational graphs in order to dynamically compute gradients. Computational graphs map out the operations and dependencies within a neural network that are transcribed into specific instructions depending on the available hardware. This enables automatic differentiation on a wide array of computational platforms, from traditional CPUs to specialised hardware such as GPUs or TPUs. This adaptability significantly speeds up the training time [18] and enables the rapid advancement of deep learning applications.

While gradient descent constitutes the foundational pillar for neural network training, limitations such as the predisposition to get stuck in local minima and the sensitivity to the learning rate have required the development of more advanced algorithms. Techniques such as Adagrad [82], Adadelta [331], Adam [149], and RMSProp introduce adaptive learning rates based on previous gradients and employ concepts like momentum for a smoother and more rapid convergence. By addressing the limitations of traditional gradient descent, these approaches improve the ability of deep neural networks to navigate complex loss functions with greater accuracy and speed.

2.2.6 CNN

Convolutional Neural Networks (CNNs), also known as Convolutional Networks (ConvNets), are a deep learning architecture that draws inspiration from the inner workings of the visual cortex. Initially conceptualised by Kunihiko Fukushima in his 1980 paper "Neocognitron: A self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position" [99], CNNs were later popularised through Yann LeCun's application in Handwritten Zip Code Recognition [168, 169]. This type of architecture specialises in processing data that present

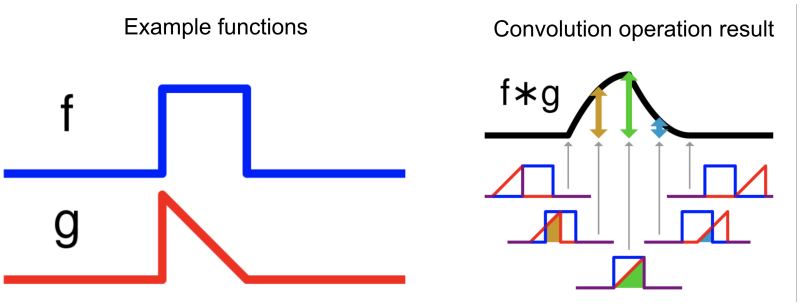


FIGURE 2.16: The mathematical operation of Convolution.

a grid-like topology, such as discrete, consistent time interval sampled time series, images, and three-dimensional volumetric data. Characterised by their use of convolutional layers, CNNs are defined by the usage of the convolution operation for at least one of their layers [106] in order to leverage spatial local hierarchies. The capacity to manage and interpret data with grid-like structures makes them exceptionally well-suited for tasks such as image and video recognition and spatial data processing.

Building upon the principles of MLPs, CNNs can be conceptually regarded as MLPs with an infinitely strong locality prior; this architecture focuses exclusively on functions that incorporate local information and interactions. This prior is enforced through the use of the convolutional operation, which endows the CNN with its powerful feature extraction.

The mathematical concept of convolution (Fig. 2.16) refers to both the process of computing it and the result. It is mathematically defined as a mathematical operation on two functions, f and g, that produces a third function, $f * g$, that equals the integral of the product of the two functions, where one of them is flipped about the y-axis and then shifted. This operation is expressed through the Equation 2.3, where the integral is evaluated for all possible values of shift. Due to the property of commutativity of the operation, the selection of the function to be shifted does not influence the result.

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{2.3}$$

In the field of Computer Science, especially in the domains of Digital Signal Processing and analysis, the operation of convolution is adapted to accommodate functions f and g defined on the set of $\mathbb{Z}$ integers. The discrete form of the convolution operation of f and g described by the Equation 2.4 and illustrated in Fig 2.17 sums over all integer points of the functions while shifting g. Considering the most common type of convolution within the context of CNNs, two-dimensional convolution, and a kernel function g that is 0 everywhere except to a finite set of points, we can express convolution as in Equation 2.5 where the output at position $[i, j]$ represents the accumulated result of applying the kernel g over the input f at each position $[i, j]$.

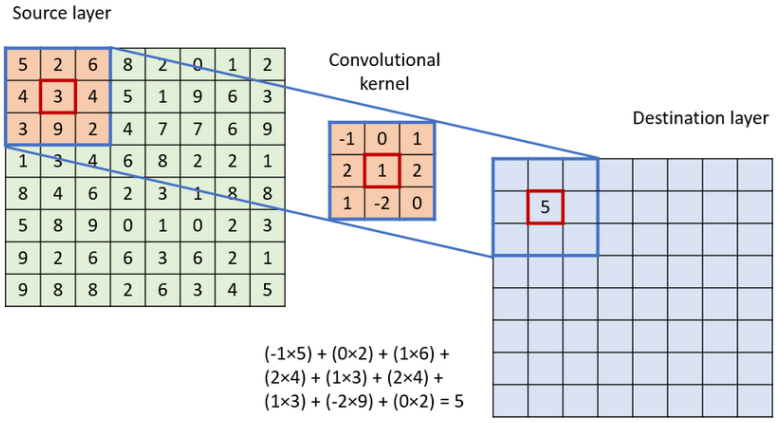


FIGURE 2.17: Discreete convolution.

This approach closely mirrors the function utilised for CNNs. In practice, the operation applied is technically the cross-correlation operation, given by the Equation 2.6. The difference between cross-correlation and convolution is that in the case of convolution, the kernel function g is flipped around the y axis, while the same is not true for the cross-correlation operator. This distinction offers the convolution operator its commutativity property. Despite this difference, the usage of "convolution" persists for historical reasons. Specifically, in the context of CNN, the distinction becomes less significant since the kernel function is learnt, and the network would be equally able to learn kernels that are flipped versions of those used in the strict mathematical definition of convolution.

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m] \tag{2.4}$$

$$(f * g)[i,j] = \sum_{m}\sum_{n} f[i-m, j-n]g[m,n] \tag{2.5}$$

$$(f * g)[i,j] = \sum_{m}\sum_{n} f[i+m, j+n]g[m,n] \tag{2.6}$$

Sparse Connectivity or Sparse Interactions (Fig. 2.18) constitute a powerful design feature that significantly improves the network's performance and efficiency. With respect to the number of parameters used, a fully connected layer demands $m \times n$ parameters to handle m inputs and n outputs, thus exhibiting a computational complexity of $O(m \times n)$ while a convolutional layer utilises kernels k with a size that is commonly several orders of magnitude smaller than m enforcing the locality of the results and reducing the parameter count to $k \times n$ and a computational complexity of $O(k \times n)$. The sparse connectivity of CNNs reduces memory usage and computational requirements, enabling the training of deeper and more complex networks.

Building on the advantages of Sparse Connectivity, CNNs leverage the concept of parameter sharing to further enhance their efficiency. Parameter sharing refers to

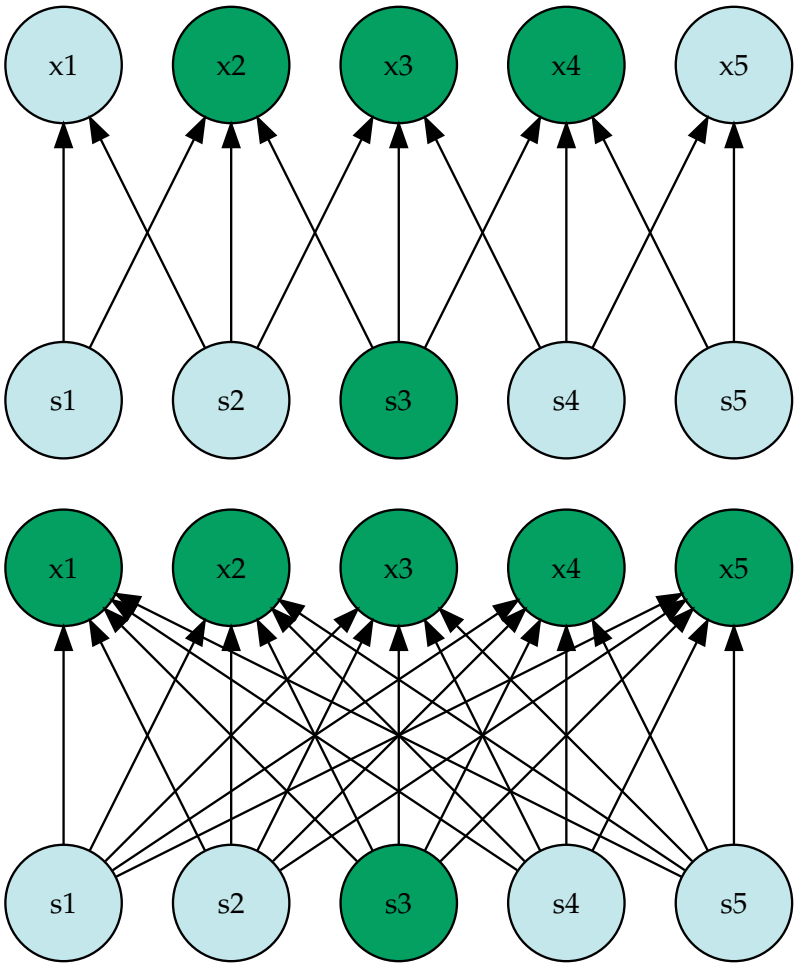


FIGURE 2.18: Sparse connectivity. One input unit x_3 is highlighted together with the output units s that are affected by it. In the case of sparse interactions (top), when s is formed by convolution with a kernel of width 3, only three outputs are affected by x_3. In the case of a Fully Connected Network (bottom), the connectivity is no longer sparse, and as a consequence, all the outputs are affected by x_3.

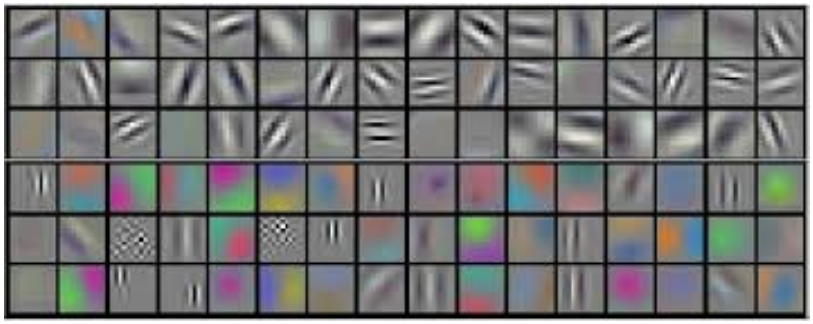


FIGURE 2.19: Visualisation of a random subset of the feature filters in a Convolutional Neural Network at the first layer.

the usage of the same parameters across different parts of the network. In CNNs, this means that each member of the kernel is applied across every location of the input (in some cases excluding the boundary pixels). This uniform application across the input ensures that the model learns a single set of parameters that can recognise a feature anywhere within the input. The repeated application of the same kernels promotes the detection of useful generic functions. Typically, the initial layers learn to detect simple universal patterns like edges and gradients, as demonstrated by Fig. 2.19. This built-in preference for generalised feature detectors contributes to the model's ability to train effectively even on smaller datasets and allows it to better generalise to new data.

The particular form of parameter sharing that CNNs employ grants the network the property of translational equivariance. This concept involves a given function space $X(\Omega)$, its transformed counterpart $X(\Omega')$ and a transformation group G with actions $\sigma : G \times X(\Omega) \rightarrow X(\Omega)$ and $\sigma' : G \times X(\Omega') \rightarrow X(\Omega')$; a function $f : X(\Omega) \rightarrow X(\Omega')$ is said to be G-equivariant if and only if it satisfies the condition shown in Equation 2.7. In essence, The property of equivariance, as graphically illustrated in Fig 2.20, refers to the capacity of a system to reflect the application of a transformation to the input results in a corresponding transformation in the output. This property is essential for computer vision tasks such as object detection, where the model is tasked to identify and locate objects regardless of their position in the input image. Embedding this prior to the task into the architecture of the model is one of the reasons for which CNNs excel in object detection.

Computer vision tasks, such as image classification, depend on a specialised form of equivariance named invariance. Invariance is very similar to equivariance, being expressed over the same set of spaces and actions. The distinction is that Invariance must satisfy the condition shown in Equation 2.8. The property of invariance dictates that the output of the function should not be changed by any of the actions σ; this is visually depicted in Fig. 2.20. Invariance to transformations such as rotation, translation and scale is essential in the task of image classification, where the identity of the class of an object should remain consistent regardless of its position, orientation or size within the image.

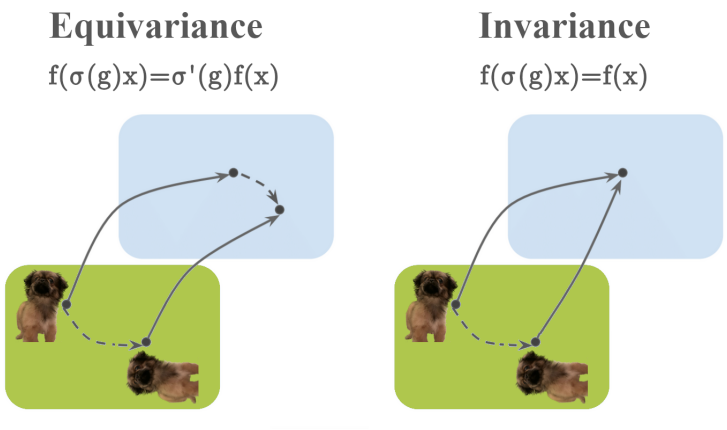


FIGURE 2.20: Graphical representation of the properties of equivariance (left) and invariance (right).

$$f(\sigma(g,x)) = \sigma'(g, f(x)) \qquad \forall g \in G \tag{2.7}$$

$$f(\sigma(g,x)) = f(x) \qquad \forall g \in G \tag{2.8}$$

CNNs achieve remarkable efficiency in processing grid-like data structures, even surpassing human-level performance in classification tasks [120], by incorporating priors such as translational equivariance and locality in the architecture of the network through operations such as convolution. While the locality prior specialises the network in dealing with images and other similar types of data, it can also limit its performance with data that presents long-range interactions such as natural language processing or the amino-acid sequences of proteins.

Another critical challenge emerges from the need for extensive datasets of annotated data to train these models effectively. The performance of the CNN is heavily dependent on the availability and quality of labelled data. This can pose a significant challenge, especially in domains where data annotation is labour-intensive and requires specialised knowledge, such as the field of medical imaging. A strategy commonly employed to tackle the lack of specialised annotated datasets is pretraining on large, generic datasets, followed by fine-tuning on smaller, domain-specific datasets. This approach leverages the transferability of learned features across potentially related tasks. The success of pretraining on out-of-distribution datasets closely depends on the intersection of overall features of the datasets employed; a high degree of similarity can facilitate the transfer of useful features, while a significant difference between the datasets might even adversely affect the model's performance.

The black-box nature inherent in deep neural networks represents another limitation of CNN models that gets compounded by the specific operation of convolution. This characteristic makes these networks particularly vulnerable to adversarial attacks, in which small alterations to an image, imperceptible to the human eye, can dramatically change the classification and associated confidence levels [46, 180].

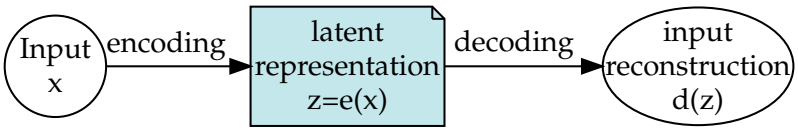


FIGURE 2.21: Workflow illustration of the conceptual functionality of an autoencoder.

Despite their limitations, CNNs have demonstrated remarkable success across a wide range of applications, particularly in computer vision tasks. They are the basis of invaluable tools in diagnosing diseases with high accuracy. For example, CNNs have shown proficiency in identifying lung squamous cell carcinoma from pathology images [165] and detecting skin and breast cancer [12]. They have even surpassed the diagnostic accuracy of trained pathologists in predicting cardiac failure from endomyocardial biopsy samples [219]. The expansion of the convolution operation to process three-dimensional data has further enabled their application in detecting emphysema in pulmonary 3D-CT images [121], brain lesions in MRI images and even in detecting binding affinity [133, 244]. These achievements illustrate the power of embedding the correct priors in the architecture of the model and establish CNNs as a fundamental architecture of deep neural networks.

2.2.7 Autoencoders

Autoencoders are an architecture of deep learning (Fig. 2.21) specialised in handling unsupervised learning tasks. They can be conceptually considered a specialisation of the multi-layer perceptron. This architecture emerged from the cumulative work of various researchers [16, 257] and popularised by Bourlard et al. in 1988 [31]. Autoencoders aim to learn efficient representations of unlabelled data. They employ a specialised architecture 2.22 or specific loss functions that restrict the internal representation into learning specific features and becoming invariant to irrelevant details. Through the process of encoding input data into a latent-space representation and subsequently reconstructing the output from the internal representation, autoencoders learn transformative features intrinsic to the data. This efficient mapping that doesn't require labelled datasets removes the requirement of annotated databases during training and helps it diversify its potential objectives, from dimensionality reduction to feature extraction and even as the basis for generative models.

Following their introduction, autoencoders have been traditionally employed for dimensionality reduction and feature learning, serving as prime candidates for extracting salient features from data without supervision. The autoencoder's ability to take advantage of unlabelled data stems from their design to encode input data x into a latent space h and reconstruct the input from the latent representation, implementing effectively an identity function $x = g(f(x))$. In order to enforce the learning of desired features, the identity function can be implemented in a different space than the one containing the input data through a transform function s; in this

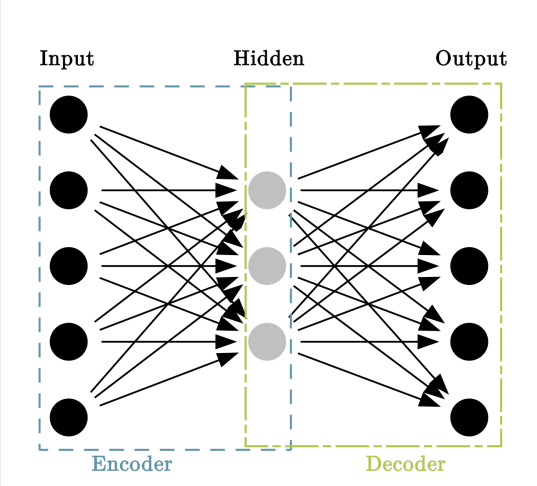


FIGURE 2.22: Undercomplete autoencoder.

case, the identity function would take the following form $s(x) = s(g(f(x)))$. This strategy is known as score matching.

The hallmark characteristic of the traditional autoencoder architecture is constituted by its distinctive "bowtie" configuration (Fig. 2.22). The model can be conceptually divided into two major functional components: an encoder $h = f(x)$ that compresses the input data into a compact latent representation and a decoder $g(h)$ tasked with reconstructing the input data from this latent representation. Learning the identity mapping $g(f(x)) = x$ with no restrictions would lead to trivial solutions, rendering the autoencoder ineffective for any practical learning. To circumvent this, autoencoders are intentionally designed with limitations that compel the model to prioritise specific representational aspects that are most crucial for reconstruction. One common approach is to design the autoencoder to have a hidden layer with a dimensionality smaller than the input space and, therefore, achieve compression into a smaller latent space; this type of restriction classifies the autoencoder as an "undercomplete" autoencoder. Other strategies involve the employment of specialised loss functions that guide the autoencoder towards learning significant, meaningful features of the data.

The mechanism of work for autoencoders exploits the idea that data tends to cluster around low-dimensional manifolds or a limited set of such structures. In their learning process, autoencoders aim to learn the underlying manifold's structure. This process is achieved through the navigation between two contrasting goals: the objective of encoding data into the latent representation h from which the original data X can be accurately reconstructed by the decoder and the constraints imposed through the architecture or loss function. This dynamic interplay between two seemingly opposing objectives provides the networks with its representational power. The autoencoders achieve a balance whereby the latent representation allows for an approximation of the data while simultaneously adhering to the constraints imposed that inhibit trivial solutions, such as duplicating the input or ignoring it

altogether. By virtue of this balancing act, the model implicitly learns a local coordinate system for the manifold it models, with the encoder mapping changes invariant to those orthogonal to the manifolds, essentially learning to ignore variations in data that do not contribute to its underlying structure.

The most common fundamental type of constraint implemented in autoencoder architectures is dimensionality reduction, a characteristic that gives rise to the undercomplete autoencoder. This version is easily recognised by its "bowtie" architecture shape (Fig. 2.22), designed to constrict the dimensions of the latent space to be smaller than those of the input space. Through this informational bottleneck, the autoencoder is forced to prioritise an embedding manifold that retains the most relevant features of the data during the encoding process. This architecture ensures meaningful data compression and feature extraction as the informational bottleneck prevents simple data replication without substantial feature learning. The number of features extracted is determined by the dimension of the bottleneck layer, and the degree of compression dictated by the ratio between the sizes of the input and hidden layer. The learning process of undercomplete autoencoders is formalised through the optimisation of a loss function $L(x, g(f(x)))$, which quantifies the dissimilarity between the input and its reconstruction; one loss function commonly employed for this is the mean squared error. This type of autoencoder can be conceptually conceived as a generalisation of Principal Component Analysis that incorporates nonlinear transformations through nonlinear activation functions. An undercomplete autoencoder with linear activation functions and a mean squared error loss function can learn to approximate Principal Component Analysis when the bottleneck layer's dimensionality is set to the number of principal components.

Another common type of constraint implemented in autoencoder architectures focuses on encouraging the model to learn meaningful features through the integration of specific autoencoder regularisers in the loss function; autoencoders employing this strategy are classified as regularised autoencoders. These autoencoders, even though they can work in collaboration with the dimensional reduction of the undercomplete autoencoders, are not strictly bound by dimensionality and capacity limitations, even allowing for architectures where the latent representation can surpass the size of the original encoding space. Regularised autoencoders introduce additional terms in the loss function to encourage desirable properties in the model, such as sparsity of activations or robustness to input corruption.

Sparse Autoencoders introduce a sparsity penalty $\Omega(h)$ to the traditional identity function the autoencoder is trained with. This forces the model to learn efficient and compact representations of the data. The loss function of the sparse autoencoder, Equ. 2.9 encourages a sparse encoding representation by incorporating a term that penalises the number of activations within the hidden layer. This penalty $\Omega(h)$, alongside the reconstruction loss, ensures that the autoencoder activates only a minimal number of neurons at any given time, prioritising the most significant features of the data. Through their pioneering research into sparse autoencoders, Ranzato et

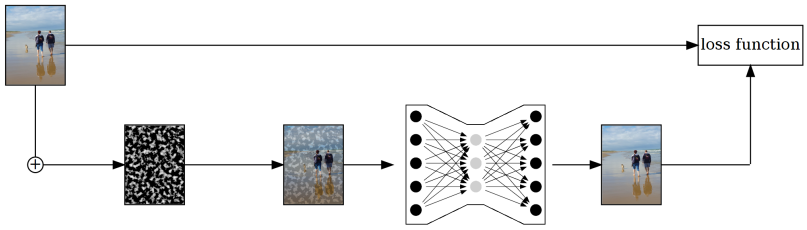


FIGURE 2.23: Denoising autoencoder.

al. [247, 248] have demonstrated the models' power to discover useful features as a part of their optimisation process. This type of autoencoder is most typically used in a pretraining stage for learning features for subsequent tasks, such as classification.

$$\begin{aligned} loss &= L(x, g(f(x))) + \Omega(h) \\ \Omega(h) &= \lambda \sum_i |h_i| \end{aligned} \tag{2.9}$$

Contractive Autoencoders, introduced by Rifai et al. [251], incorporate a regularisation term that penalises the sensitivity of the hidden representation to slight variations of the input. The regularisation term $\Omega(h)$ (equation Equ. 2.10)) represent the squared Frobenius norm of the Jacobina matrix of the encoder function's partial derivatives with respect to the input. This term forces the autoencoder to learn a mapping that is robust to small changes in the input data, effectively learning features that are invariant to minor perturbations. The essence of contractive autoencoders lies in their training to resist slight input perturbations, encouraging the model to contract a neighbourhood of points. One significant challenge with contractive autoencoders is the computational expense of calculating the loss function for deeper networks. To reduce the computational cost of training, Rifai et al. [251] suggest training a series of single-layer autoencoders sequentially, with each layer trained to reconstruct the hidden representation of the previous layer; while this is not identical to applying a contraction penalty over the entire deep model, it retains many of the desirable contractive properties.

$$\begin{aligned} loss &= L(x, g(f(x))) + \Omega(h) \\ \Omega(h) &= \lambda \sum_i ||\nabla_x h_i||^2 \end{aligned} \tag{2.10}$$

Denoising Autoencoders alter the traditional autoencoder learning objective; Vincent et al. [308] introduce a novel approach to learning useful features without adding an explicit regularisation term to the loss function. The denoising autoencoder works by adding noise to the input x before processing it through the network. Specifically, the model takes a corrupted version of the input and learns to reconstruct the original, uncorrupted input x as described by equation Equ. 2.11 and illustrated by figure 2.23. This technique, as detailed by Bengio et al. [21] and Alain and Bengio [7],

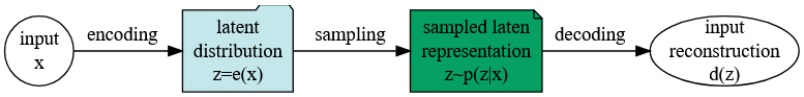


FIGURE 2.24: Variational Autoencoder.

demonstrates that denoising autoencoders can learn the structure of the data as a natural result of minimising the reconstruction error.

$$\begin{aligned} loss &= L(x, g(f(\tilde{x}))) \\ \tilde{x} &= x + noise \end{aligned} \tag{2.11}$$

Variational Autoencoders, as introduced by Kingma et al. [150] and Rezende et al. [250], mark a significant advancement in autoencoder technology by framing the autoencoder architecture within a probabilistic generative model framework. This innovative approach distinguishes variational autoencoders from traditional autoencoders by choosing to bypass the classical encoding of the input data into a single point in the latent space with the parameters of a probability distribution (usually the mean and covariance). This encoding strategy allows the VAE to generate a distribution over possible latent variables and, as a result, to operate as probabilistic generative models that can be fully trained using gradient-based optimisation methods. The decoder has the same functionality as the classical autoencoder, with the mention that before decoding, the probability distribution is sampled, and the result is what is processed by the decoder, as illustrated by Fig. 2.24. Through the incorporation of probabilistic inference principles, the latent space is systematically guided to exhibit a well-organised, continuous and smooth structure. This property of the latent space enables the use of the Variational autoencoders for the generation of complex, high-dimensional data.

Expanding beyond their original use, modern autoencoders have evolved beyond their deterministic function to encompass stochastic mappings, characterised by $P_{\text{encoder}}(h|x)$ and $P_{\text{decoder}}(x|h)$, marking a transition from deterministic to generative models capable of capturing complex data distributions. This evolution marks the autoencoders not only as feature extractors but also as versatile models for generating new data.

Autoencoders share common challenges with deep neural networks, such as the necessity for substantial amounts of data and computational resources. Notably, an advantage of autoencoders lies in the ability to train on unlabelled data, reducing part of the data acquisition challenge. Autoencoders also present novel challenges. Among these is the phenomenon of mode collapse, where the model, if lacking representational capacity, might default to predicting the average of the training dataset rather than capturing the underlying structure of the data. This issue underscores the importance of carefully choosing the network's architecture and hyperparameters to avoid undermining the models' ability to learn meaningful representations.

Autoencoders have proven to be incredibly versatile, finding applications across a broad range of domains that underscore their utility and adaptability. In semantic hashing, autoencoders have been employed to create compact representations for text and images, facilitating efficient information retrieval and similarity search, as demonstrated in the works of Salakhutdinov and Hinton [262] in the case of text and Krizhevsky and Hinton [156] for images. The domain of bioinformatics has particularly benefitted from the capabilities of autoencoders. Variational autoencoders have been instrumental in obtaining continuous latent spaces conducive to the generation of new molecules through operations such as interpolation [113], measuring molecular similarity through measuring the distance of latent space representations [263] and encoding the 3D atomic structures in translation-, rotation-, and permutation-invariant latent representations [218].

Beyond these applications, autoencoders have also been used in protein function prediction [71], modelling protein-to-protein interactions [65], predicting drug responses from genomic data [53] and molecular fingerprint representations [261], showcasing their potential role in accelerating discovery in the pharmaceutical and biomedical fields. The adaptability of autoencoders extends to anomaly detection [217], where they excel in identifying outliers or novel phenomena within large datasets. Denoising autoencoders have been utilised to gather insights into complex biological processes, such as microbe-host interactions from gene expressions [291], further emphasising the board applicability and transformative potential of autoencoders in modern deep learning applications.

2.2.8 Transformer

The Transformer architecture, introduced by Vaswani et al. [303], (Fig. 2.26) represents a paradigm shift in deep learning. Originally developed as a sequence-to-sequence model attempting to tackle the complex task of machine translation, the Transformer quickly demonstrated its efficacy across a broad spectrum of domains [134]. Its design features an autoencoder meta-structure where both the encoder and decoder components consist of repeated subunits named transformer blocks, a component designed to complement the modelling capabilities of the attention mechanism. The attention mechanism is pivotal, enabling the model to dynamically allocate focus to different segments of the input, allowing the model to learn the underlying structure of the data and model dependencies irrelevant to their separation. This ability to understand long-range dependencies has proven essential for fields such as natural language processing and bioinformatics.

Prior to the development of the transformer, long-range dependencies were modelled with the help of recursive neural network architectures, such as recurrent neural networks and, later, long short-term memory networks [123]. These models often struggled to capture relationships between elements separated by long distances within the input sequence. One of the main reasons recursive models struggle to map long-range interactions is the problem of vanishing gradients, which impedes

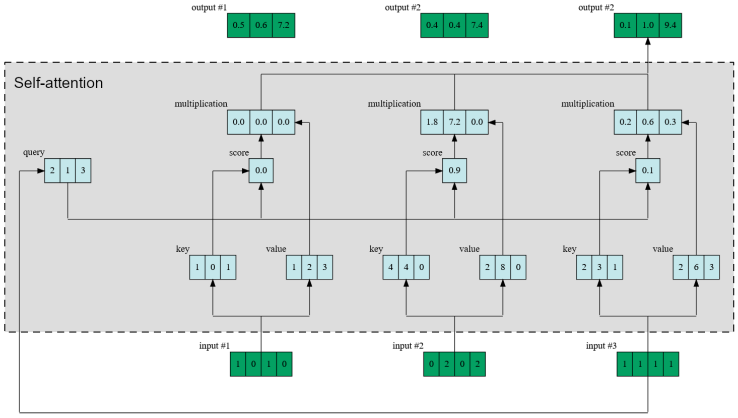


FIGURE 2.25: Internal workflow diagram of the self-attention mechanism.

the model from retaining information over extended sequences. Another limitation of these models is the inherently sequential evaluation of the model due to the model's internal state getting updated with the processing of every new token and, therefore, being dependent on the output results of the previous iteration. In contrast, the Transformer architecture overcomes these limitations through its innovative design, which eliminates the need for sequential data processing on the encoder side. This is achieved by leveraging an attention mechanism that allows each token to directly interact with every other token, capturing dependencies without the constraints imposed by the structure or recursive models. Overcoming the sequential requirement of previous approaches allows the transformer to enhance its efficiency through the usage of massively parallel computation units such as graphical processing units and tensor processing units. The transformer's ability to handle variable-length input sequences further adds to its versatility, making it adept at processing a wide range of data types. This combination of parallelisability, flexibility, and capacity to model long-range dependencies are some of the reasons transformers are the foundation of many of the state-of-the-art solutions in natural language processing, image processing, signals processing, and bioinformatics.

At the heart of the Transformer model lies the multi-head scaled dot-product attention, a specific iteration of the broader deep learning technique known as attention. This component is pivotal in enabling the model to selectively focus on different parts of the input data. This attention mechanism operates on a simple yet powerful principle, using three different vectors named key, query and value, derived from the input sequence; the key and the query are used to determine the distribution of attention over the components of the input sequence; the value is then weighted according to this distribution to produce the output of the attention mechanism.

In order to compute the attention, the mechanism follows a relatively simple process. Given a vector of input values $h_1, h_2, \cdots, h_N \in \mathbb{R}^{d_1}$ with d_1 denoting the

dimensionality of the input data, three derived vector values of dimensionality d are computed key k, query q and value v. The next step involves the calculation of attention scores $e_i = \frac{q^T h_i}{\sqrt{d}}$ by multiplying the q value of the current token with each of the k values of the other tokens in the sequence, resulting in a value e_i that quantifies the strength of the interaction between the two tokens. The product of q and q is divided by the square root of the dimensionality d in order to help stabilise the gradients during training. The attention distribution is computed from the attention scores e_i using the softmax function (Table 2.2). This distribution is utilised to perform a weighted sum of the values, leading to the final output $a = \sum_{i=1}^{N} a_i v_i$ of the attention mechanism. The computational graph for computing self-attention is illustrated in Fig. 2.25. A more detailed exploration of the attention mechanism and its application in deep learning is provided by Niu et al. in their "A review on the attention mechanism of deep learning" work [220].

One limitation of the attention mechanism is constituted by its capacity to model a single aspect of the data through its attention distribution, which could severely restrict the model's ability to handle complex data with multi-faceted interactions. To overcome this limitation, multiple attention mechanisms are employed in parallel, forming the multi-head attention mechanism. This approach allows each attention head to focus on a different facet of the input and, by concatenating all of the results from the different attention heads, enables the model to capture a wide range of interactions and relationships within the data.

To illustrate the application of the transformer architecture, a concrete example from the transformer's original problem domain, machine translation. The process begins with splitting the initial input into tokens, common words or groups of characters for which a fixed-length vector representation named word embedding has been previously learnt. Word embeddings condense the information from the discrete, sparsely populated space of the words of a language into a denser vector representation that supports mathematical operations. This is done in order to increase the processing efficiency through the reduction of dimensionality and to facilitate the learning process through the support of mathematical operations such as addition and subtraction. The embedded vectorial representation allows for operations such as king $-$ man $+$ woman $=$ queen. Through the usage of different embedding processes, multimodal models can be achieved, models capable of handling different types of input data such as images, point clouds, and vidoes. Due to the fact that the transformer architecture has no embedded sequence prior, at this stage, a positional encoding is added to the word embeddings to offer the model the capacity to pay attention to the sequential ordering of over two million tokens [75].

The generated embeddings are passed to the encoder component of the transformer, which is responsible for encoding the input sequence into an informative, high-dimensional representation. The encoder employs the attention mechanism in order to learn semantic interactions between the embeddings and generate new

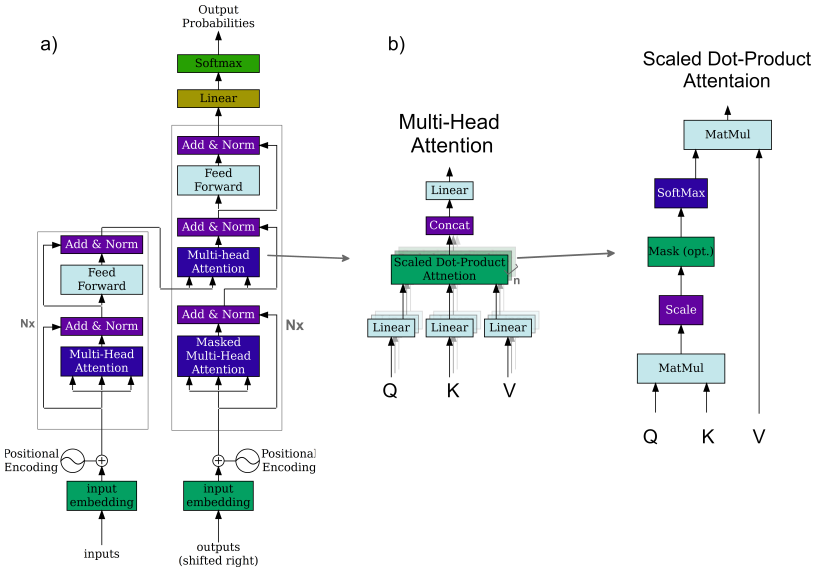


FIGURE 2.26: Graphical illustration of the Transformer model.

representations that capture contextual information. Transformer models usually employ multiple transformer blocks, allowing the iterative refinement of the embeddings to include deeper, more nuanced relationships. Each block features a multi-head normalised attention layer and a fully connected feed-forwards network, augmented by normalisation layers and residual connections. These components work to stabilise the training process and enhance the gradient flow, ensuring that each word's vector representation captures rich semantic detail.

The decoder component works in symmetry with the encoder, with the role of bringing the high-dimensional representation back into the natural language space. Its structure closely resembles that of the encoder, with the addition of one cross-attention mechanism positioned after the self-attention mechanism. Cross-attention works similarly to self-attention with the distinction that the v value and k key are generated from the output of the corresponding transformer block in the encoder, facilitating the nuanced interplay between the input sequence being processed and the emerging output sequence at each stage. The output text is generated sequentially in an autoregressive manner, where the prediction of each subsequent token is predicated upon those already generated. This iterative process begins with the decoder ingesting a special embedding token '<start>'. The input then traverses the decoder's layers, capturing not only semantic information from the current sequence but also incorporating information from the encoder. The output of the last decoder transformer block is fed into a multi-layer perceptron; its output is fitted to a probability distribution across all the embedding tokens. From this distribution, the next word of the sequence is selected, usually dictated by a hyper-parameter called temperature that introduces a little variability in the output through the random proportional selection from among the top percentile of probabilities. The selected embedding is transformed back into the natural language space and appended to the input sequence. The process of ingesting, processing and selecting a word from

the resulting probability distribution is repeated until the special embedding token '<stop>' is generated or a specific number of iterations has been achieved.

The transformer inherits the limitations of deep neural networks, including the black-box nature of its decision-making processes, which presents challenges to interpretability, making it difficult to understand the model's reasoning and hindering the adoption of this type of model in fields that require explainability, traceability and transparency. Another significant limitation of the transformer architecture is imposed by its requirement for significant processing power, especially from graphical and tensor processing units, coupled with considerable memory consumption. The attention mechanism, which lies at the heart of the transformer architecture, results in memory requirements that scale quadratically with sequence length. Improving the attention mechanism on long sequences [56] as well as improving the memory efficiency [52] are current directions of study in the field, particularly since increases in model size have shown to improve performance [145].

The transformer architecture forms the basis of many current state-of-the-art models across various domains, showcasing its versatility and efficacy. In natural language processing, it has been pivotal in advancements in machine translation [88, 281, 303, 343], summarisation [19, 243], question answering [184, 243], text classification [243] and text generation [243], including the development of the GPT series [242]. The introduction of the Vision Transformer (ViT) [80] makes the introduction of the architecture into Computer Vision, where it revolutionised tasks like classification [80, 177, 185, 340], detection [42, 50, 107, 185, 334], image segmentation [78, 285, 321]. In the subfield of medical images, state-of-the-art models for image segmentation [320] and image analysis [186], the transformer architecture outperforms existing methods. In the realm of Bioinformatics, transformers have been instrumental in predicting protein folding [144, 325], predicting protein function [41], predicting drug-target [81, 130] and drug-drug interactions [148, 271], classifying and identifying viral sequences [258], secondary structure prediction [90], and solvent accessibility [131]. This broad array of applications underscores the transformative impact of the Transformer architecture across disciplines, establishing it as a cornerstone architecture of modern machine learning.

2.2.9 Large Language Models

Large Language Models (LLMs), also known as foundational models, represent a significant engineering advancement in the field of machine learning. They are massive models that predominantly employ the transformer architecture in order to excel at general-purpose language generation and a variety of other natural language processing tasks, such as text summarisation and question answering. The performance of LLMs stems from their training on internet-scale amounts of data. This extensive training, coupled with their apparently insatiable representational power, gives birth to emergent sophisticated cognition capabilities such as mathematics and reasoning, making them highly versatile tools.

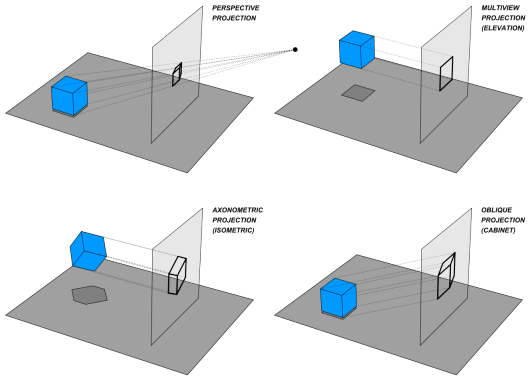


FIGURE 2.27: Different 2D plane projections of an unknown 3D object (cube in this instance).

This type of model exhibits an unprecedented level of artificial general intelligence [40], reaching performances that are beginning to approach and even surpass the average human level in certain domains [40]. This remarkable achievement can be better contextualised by comparing biological and artificial neural constructs: the human brain consists of approximately 86 billion neurons; in comparison, chat GPT-3, a landmark in LLM development, is constituted of 175 billion parameters [39] and has been trained on a corpus exceeding 500 billion words [39]. This comparison highlights the scale of modern models and the complex architecture that underlines the sophisticated processing capabilities.

The success of LLMs can be partially attributed to their foundation in the Transformer architecture, which seems to exhibit a linear tendency to improve performance by increasing the model size. [145]. This scaling law suggests that LLMs benefit from an 'economy of scale' where there is a direct relationship between increased computational resources, larger amounts of data, and model performance. This data-driven scalability capacity of the LLM is enabled by the ability of Deep Learning architectures to autonomously generate, select and transform data features. As a result, these models succeed in effectively leveraging vast quantities of data, with the help of immense computational power, into advanced language understanding and generation capabilities.

In order to enable the training on internet scale amounts of unstructured data, LLMs leverage more complex training strategies. These models are commonly pretrained, employing unsupervised learning techniques such as next-word prediction or masked word prediction, where the model attempts to predict missing or subsequent words in a sentence. This forms the foundational learning phase, in which the model learns the underlying structure of language. During this phase, as a side-effect of its training objective, the model accumulates a statistical understanding of the real world. This mechanism of understanding the real world as a byproduct of the next-word prediction goal can be conceptually compared to learning to predict two-dimensional projections from a set of similar projections of an unknown three-dimensional object (Fig. 2.27), where the word representations constitute the

two-dimensional representations and the real-world concept is represented by the hidden three-dimensional object. Even though the objective does not explicitly require knowledge of the hidden object, a conceptual representation of it is required to achieve good predictive performance. Subsequent to this unsupervised phase, the models are often fine-tuned either on smaller labelled datasets or using reinforcement learning [283] such that the model's response becomes increasingly aligned with human-like understanding and to the format of its task.

Even though they exhibit behaviours similar to human knowledge and cognition, LLMs differ fundamentally in their approach to learning and utilising information. Unlike human cognitive processes that acquire and apply factual knowledge from direct experiences and education, LLMs are essentially supercharged autocomplete systems. Their training does not focus on the facts of the world but on next-word prediction. This means that their grasp of logic and factual information is an emergent feature of their primary task [312]. Their training goal relies on capturing the patterns and statistical distributions of words within vast training corpora. For instance, when given the prompt, "The first antibiotic discovered by Alexander Fleming was," an LLM is likely to complete the statement with "penicillin". Even though this is the correct answer and is the same as the expected answer from a human, this response is not derived from an internal database of verified facts but from statistical likelihoods based on the training data. This mode of operation makes LLMs prone to propagating common misconceptions and creates a misalignment with human communication. While humans typically use factual knowledge as a basis for discussions, LLMs operate purely on the frequency and relationships of words and phrases within their training corpora. This distinction underscores a critical difference in how LLMs and humans represent and use knowledge, posing challenges for applications that require precise and reliable factual information.

Another consequence of the training process is the phenomenon of "hallucinations" [140], where models generate incorrect or misleading information with authority and confidence. As probabilistic models, LLMs tend to fill gaps with plausible but potentially inaccurate information, which can mislead users who rely on their outputs for decision-making or factual accuracy. Additionally, the risk of "model collapse" [278] is a growing concern as LLMs are trained on ever-expanding internet-scale datasets that increasingly include machine-generated or manipulated data. This issue has the potential to lead to worse-performing future models, as demonstrated by Bohaceck et al. [28] and Bertrand et al. [23]. This issue is further accentuated by the fact that LLMs contain a fixed knowledge base that requires retraining in order to be kept up to date with facts and events. Furthermore, the computational and financial costs of training such models are extraordinarily high; for example, GPT-3 training involved resources equivalent to 285,000 CPUs and 10,000 GPUs, amounting to approximately 12 million dollars [341]. Furthermore, the immense datasets required for training impose additional challenges, including the reliance on heuristic methods for data validation due to the massive amount of data

that makes manual quality checks impossible, as well as the issues of copyright and personal information. The sheer amount of data poses additional issues, such as possible benchmark data contamination and the challenges of removing close duplicates. These limitations highlight the trade-offs involved in the development of LLMs and accentuate the need for careful consideration of their application context and potential risks.

LLMs represent a breakthrough in the application of machine-learning technologies across various domains. Significant advancements in conversational AI have been achieved through models such as Meta's LLaMA [297], Google's PaLM [57], BERT [70] and OpenAI's ChatGPT [39, 221], with the latter being one of the fastest most widespread publicly adopted applications. These models showcase the potent capabilities of LLMs when tailored through prompt engineering for specific tasks like dialogue, question answering, and text summarisation.

In the field of bioinformatics, specialised LLMs such as GeneGPT [142], BioGPT [188], and BioMedLM [29] have demonstrated how classical LLMs can be leveraged to mine biomedical text and anchor the model in ground truth using API calls. Models such as DNABert [139], DNABert-2 [342], and DNAGPT [332] have been used for tasks such as DNA-protein interaction, predicting cis-regulatory regions and predicting genome-wide variant effects. The field of proteomics has benefited from LLM applications such as MSA Transformer [249], ProtTrans [84], ProGen [191], ProtGPT2 [91], ProteinBERT [34] and ProLLaMA [190] for tasks such as predicting secondary structure, predicting protein function, predicting biophysical properties and predicting protein-protein interaction. These applications underscore the impact of LLMs across various scientific domains and demonstrate the capabilities of these models to tackle complex challenges.

2.2.10 Molecular Representations

Invariant to the specific machine learning task, the first stage in any computational model involves computing some form of representation for the molecular reality. Given the complex and multifaceted nature of molecules, a wide array of molecular representations exist. These range from high-resolution methods, such as the all-atom models used in molecular simulations, to geometric projections [267].

Current high-performing representations can be divided into four subcategories: Molecular Fingerprints, Voxelised Image Representations, Molecular Graphs, and learned Representations. Another widely used text-based descriptors system is SMILES, which provides a linear representation of molecular structures. Each of these approaches offers unique advantages and challenges, making them suitable for different types of computational tasks in the realm of bioinformatics.

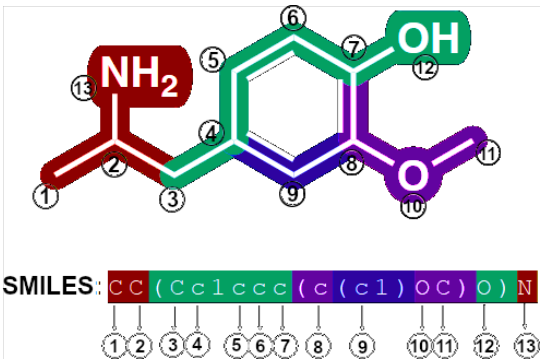


FIGURE 2.28: Illustration of the mapping from chemical structure to SMILES.

SMILES

The simplified molecular-input line-entry system (SMILES) is a widely-used text-based descriptor for molecular structure, specifically developed to store chemical data in a computer-readable format [313, 314].

In SMILES strings, atoms are represented by their chemical symbols, and aromatic atoms are denoted in lowercase. Special characters are employed to fully specify the chemical structure of a molecule following specific rules. For instance, double and triple bonds are represented by '=' and '#', respectively. Chirality is indicated using the '@' symbol, while double bond configurations are denoted by '\' and '/'. Branches in the chemical structure are marked using parentheses '(' and ')'. Cyclic substructures are encoded by breaking a single bond in the ring and labelling the matching atoms with numbers (Fig. 2.28).

Following these rules, a SMILES string is constructed by traversing the nodes of the molecular graph. Depending on the choice of starting node and traversal route, there can be multiple valid SMILES representations for a single molecule, especially in the case of large molecules. To address this, a deterministic algorithm is used to choose the starting node and traversal route, resulting in a unique SMILES representation for a molecule known as the 'canonical' SMILES [314].

Despite its utility, SMILES can encode invalid molecules, which is a potential limitation. However, the text-based nature of SMIELS and its expressiveness in encoding the molecular graph alongside stereochemistry have contributed to its widespread use in storing chemical data. The vast majority of molecular datasets used in machine learning models represent molecules as SMILES strings.

Molecular Fingerprints

The majority of classical machine learning methods require a fixed-length vector representation of molecules; these are known as molecular encodings or molecular fingerprints. The molecular fingerprints can vary widely in shape and size, providing flexibility in how molecular information is encoded.

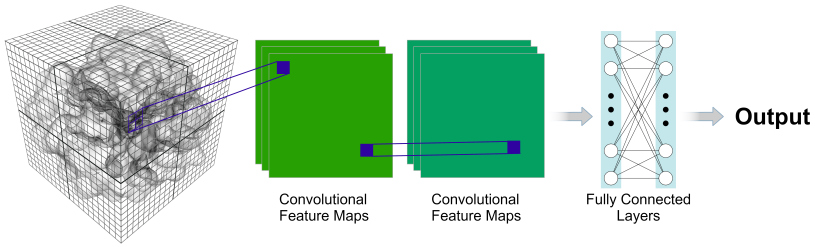


FIGURE 2.29: Schematic overview of the architecture of a 3D CNN model for protein-encoding used in protein-ligand binding affinity prediction.

One of the most widely known fingerprints is the Morgan fingerprint [210], which encodes topological information about the molecule. Extended Connectivity Fingerprints (ECFP) are another popular example, designed to capture detailed topological information about the ligand structure [252].

Fingerprints can also include 3D protein-ligand contact information. Examples of these include the Protein-Ligdan Extended Connectivity fingerprint (PLEC) [318] and Extended Connectivity Interaction Features (ECIF) [289], which capture detailed interaction information between proteins and ligands.

Molecular fingerprints are also commonly used in protein-ligand affinity scoring functions, particularly in models based on Random Forests or Boosted decision trees, due to their simplicity and explainability [206, 289, 318]. Molecular fingerprints have been employed in Graph Neural Networks as well [206].

Despite their representational flexibility, there are limitations to how much information can be encoded in a given fingerprint size. This can impact the effectiveness of the representation of complex molecules.

Voxelised Image Representation

The advancements in the field of image processing brought forth by the applications of CNNs have been extended to the field of protein-ligand affinity prediction. In this context, RGB channels, traditionally used to represent colour in images, are repurposed to encode various atomic properties such as hydrophobicity, ionizability, aromaticity, and hydrogen bond donor or acceptor properties. This innovative approach leverages the power of CNNs to process complex molecular interactions in a manner analogous to image recognition.

To apply this technique, a 3D voxel grid is created for the protein-ligand complex. Each voxel within this grid represents a specific region of the molecular structure (Fig. 2.29). This voxelised representation allows the CNN to analyse the spatial distribution and interactions of these properties within the protein-ligand complex, facilitating the prediction of binding affinities.

Substantial progress has been made in developing CNN-based scoring functions

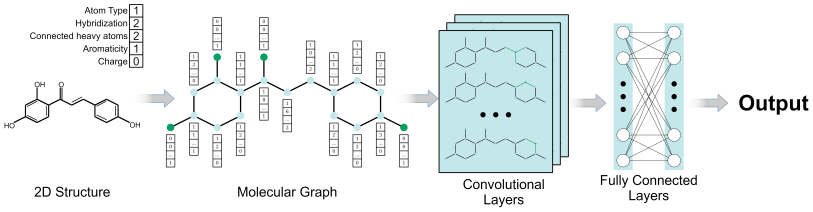


FIGURE 2.30: Diagram illustration of a ligand-based graph and setup of a Graph Convolutional Network for protein-ligand binding affinity prediction.

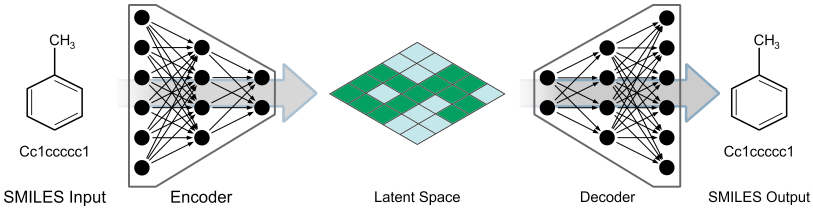


FIGURE 2.31: Autoencoder learned representation from SMILES notation.

for various applications within this domain. For instance, CNNs have been effectively employed in virtual screening [244], in the protein-ligand binding affinity prediction [141] and have been developed as scoring functions in protein-ligand docking [198].

One significant advantage of using CNNs for voxelised image representations is their ability to learn higher-order features from the training set. This ability, coupled with the inherent locality imposed by the convolutional operation, allows the model to capture complex patterns and interactions within the protein-ligand complex that might be missed by traditional methods. As a result, CNN-based approaches offer a powerful tool for predicting molecular interactions.

Molecular Graphs

Molecular graphs provide a powerful representation for encoding molecular structures in deep learning models by directly mapping atoms and bonds as nodes and edges in a graph (Fig. 2.30). This method captures the intricate connectivity and chemical relationships inherent in molecules, facilitating accurate predictions and analyses. Prominent models leveraging this approach include Graph Isomorphism Networks (GIN) [322], Graph SAGE [115] and Graph Attention Networks (GATNet) [304]. These models demonstrate the transformative potential of graph-based deep learning in bioinformatics, offering new insights and capabilities for molecular analysis.

Learned Representations

Learned representations have been developed to enable continuous space representations of molecules, offering an alternative to traditional discrete molecular representations. This approach has gained prominence due to its ability to capture

complex relationships within molecular structures, facilitating more sophisticated analyses and predictions.

Autoencoders are increasingly popular in generating molecular representations, often surpassing their graph-based counterparts in terms of usage and application. These models encode molecular data into a latent space, allowing for more flexible and powerful manipulation of the molecular features.

Learned Representations are used to enhance and complement the other representational methods. One drawback of the learned representations is that they themselves need a computer-friendly representation from which to start. Their power lies in transforming the initial representation into a more detailed and packed with essential information encoding. Models such as word2vec [202] and mol2vec [137] serve as two examples of the modelling power of the learned representations.

An advantage of using learned representations, particularly those derived from variational autoencoders, lies in the continuous representation of the latent space. This continuous representation is well-suited for generating new molecules, as it allows for smooth interpolation between known points and even extrapolation within the manifold space, facilitating the discovery of novel compounds with optimised properties.

Molecular representations are essential for bridging the gap between complex molecular structures and machine learning models. Each representation method, SMIES, molecular fingerprints, voxelised images, molecular graphs, and learned representations, offers unique strengths and trade-offs, making them suitable for different bioinformatics applications.

Chapter 3

Protein Folding, AlphaFold

This chapter delves into the complexities of protein folding from the perspective of AlphaFold, highlighting its impact on protein structure prediction. It extends the work published in "An Overview of Alphafold's Breakthrough" [196] and analyses in detail the methodological advancements and computational innovations of the AlphaFold models. The chapter concludes with an exploration of the technical challenges and possible future directions.

3.1 Background on Protein Folding

3.1.1 Importance of protein Folding

Protein folding is a fundamental process in biology, where the linear sequence of amino acids in a polypeptide chain folds into a unique three-dimensional structure essential for its biological function. The biological activity of proteins, vital for nearly all cellular functions, is intrinsically tied to their three-dimensional structure [63]. This relationship is critical; any deviation from a protein's native conformation can severely affect its functionality, leading to a range of effects from loss of function to rendering the protein toxic [138], perturbing the complex balance maintained in cellular environments.

The three-dimensional structure of a protein is determined solely by its amino acid sequence [138]. This sequence-specific folding process is guided by the physicochemical properties of the composing amino acids and the physicochemical properties of the environment. Despite the complexity and variability of protein structures, the information required to fold these structures is encoded entirely within the amino acid sequence itself, independent of the synthesis mechanisms or the presence of molecular chaperones [138].

An illustrative example of the self-sufficiency of amino acid sequences in guiding protein folding is observed in the phenomenon of protein renaturation (depicted in Fig. 3.1). Experiments have demonstrated that denatured proteins, proteins which have lost all of their ordered structure due to disruptive conditions, are capable of refolding into their functional forms upon the return to normal conditions. This refolding restores virtually all of the protein's biological activity, demonstrating that

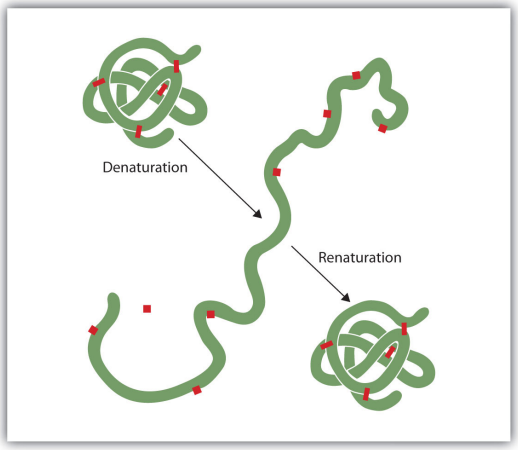


FIGURE 3.1: Illustrative depiction of the process of denaturation (the process in which a protein looses its 3D conformation) and renaturation (a protein assembling itself back in the correct fold from the denatured state).

the sequence of a protein holds all the required information to achieve the correct folding structure.

The implications of protein folding are vast, influencing numerous fields, from genetics to biotechnology and medicine. Understanding and manipulating this process can lead to breakthroughs in drug design, synthetic biology, and the treatment of diseases caused by folding abnormalities. The study of protein folding is not only crucial for understanding fundamental biological processes but also for harnessing these processes for innovative applications.

The initial direction of the research is aimed at exploring how modern machine-learning approaches could be applied to accurately determine the three-dimensional conformation of proteins based on their amino acid sequences. The success of AlphaFold's model at CASP13 has validated the foundational hypothesis that deep learning has the ability to increase the accuracy of predictive models beyond traditional approaches. The approach of AlphaFold, and later AlphaFold 2, have also validated personal intuitions regarding the use of techniques inspired by natural language processing. However, the unprecedented precision of the two models has necessitated a strategic shift in the direction of research. It has also highlighted the need for accessibility and availability of cutting-edge tools within the domain of bioinformatics.

3.1.2 Complexity of protein Folding

The process of protein folding, while fundamentally guided by the amino acid sequence, presents substantial computational challenges due to the intricate nature of protein structure and the vastness of the conformational space. Two of the main obstacles in the computation of protein folding are the energy function problem and the sampling problem.

The Energy Function Problem

One significant challenge in predicting protein folding is the accurate computation of the energy function. There is currently no straightforward method to calculate the exact electrostatic potential or the energies of bonds that are slightly stretched or in non-ideal configurations without resorting to resource-intensive quantum mechanics simulations. Precision in these calculations is crucial because even minor errors can cascade and accumulate over time, potentially resulting in a completely different fold. This issue is analogous to the N body problem in physics, where N represents the number of particles interacting with each other. For proteins, N can easily reach into the thousands, reflecting the complexity of their structures. A visual representation of this complexity can be seen in the structure of human haemoglobin (Fig. 3.2).

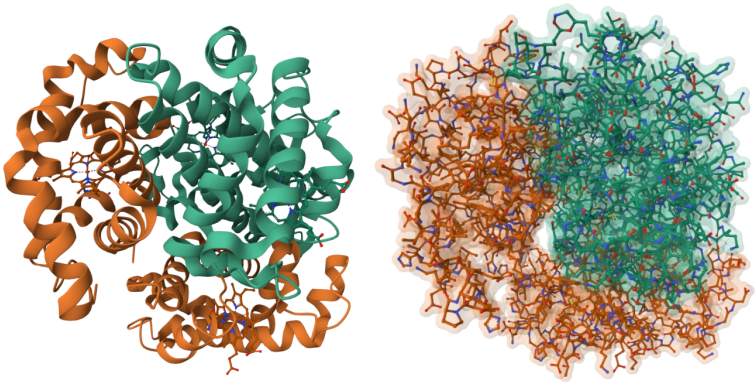


FIGURE 3.2: Example showcasing the conformation of haemoglobin (1HCO), with a length of 141 amino acids, is illustrated above. Left uses the ribbon representation, putting the accent on the 2nd order structure; Right uses the ball and stick representation, evidentiating the 1st order structure and its complexity.

The Sampling Problem

Another major obstacle in protein folding prediction is the vastness of the sample space. Even with simplifying assumptions such as bonds being of constant length, the number of possible conformations remains astronomically high. For instance, consider a protein with a modest length of 50 amino acids. The median length of a human protein is approximately 375 amino acids [37]. If each amino acid in the 50-amino-acid protein has just two rotatable bonds in the backbone, and we consider only 10-degree increments for each bond, we end up with 10^{72} possible conformations. To put this in perspective, the number of protons in the known, observable universe is approximately 1.57×10^{79} [315]. This immense number of potential conformations demonstrates the difficulty of sampling the entire conformational space to find the one with the lowest energy state.

These obstacles highlight the intricate nature of protein folding and the significant computational efforts required to predict protein structures accurately. Understanding and overcoming these challenges are critical for advancements in structural biology, drug design, and related fields.

3.1.3 CASP and PDB

The Critical Assessment of Structure Prediction (CASP) and the Protein Data Bank (PDB) are essential to the recent breakthroughs witnessed in the field of protein folding. These resources are indispensable as they enable the current advancements in structural biology by offering an impartial metric of success and the necessary data for training statistical models, respectively.

Critical Assessment of Structure Prediction (CASP)

CASP is a community-driven competition aimed at determining the state of the art in modelling protein structures based on their amino acid sequences. It is organised by the Protein Structure Prediction Center and held biannually. The contest employs a double-blind procedure, where participants submit predicting structures for proteins whose experimental structures are known to a select group but not publicly available [158]. The submissions are evaluated by independent assessors to ensure objectivity.

Scoring in CASP employs the Global Distance Test (GDT), which has been a staple metric since CASP 4. GDT assesses the quality of fold predictions by identifying the largest subset of model residues that align within a specific distance threshold to the corresponding residues in the reference structure [48]. The Global Distance Test Total Score (GDT_TS) averages these scores across cutoff points of 1, 2, 4, and 8 Å, with scores ranging from 0 (complete mismatch) to 100 (perfect superimposition). One limitation of this scoring system is that the magnitude of scores can depend on the evaluated proteins' size [336].

The Protein Data Bank

The Protein Data Bank stands as the singular global archive for structural data of biological macromolecules [22]. As a leading resource, the PDB provides access to 3D structural data of large biological molecules. These are essential for various research and development applications in molecular biology [17].

The efforts to understand and manipulate protein folding, supported by CASP and PDB, reinforce the challenge's significance in structural biology and its implications across genetics, medicine, and biotechnology [138, 197].

3.1.4 Fold Prediction Tecniques

The challenge of predicting protein folding has been approached through various techniques, each leveraging different aspects of computational biology and machine learning. These techniques aim to decipher the complex mechanism of protein folding by employing physics-based models, fragment assembly strategies, and advanced machine-learning algorithms.

Physics Simulations

Physics simulations, one of the earliest methods of predicting protein folding, utilise principles of quantum mechanics through molecular dynamics simulations. This approach is particularly effective for proteins of limited size, as it can accurately predict folding for molecules up to hundreds of residues. However, the computation cost associated with these simulations increases exponentially with the length of the protein chain. One limitation of this approach is the fact that these simulations are susceptible to inaccuracies due to potential errors in the force fields used. These errors can misrepresent the protein's behaviour under various physiological conditions. Despite their limitations, physics simulations are valued for their ability to potentially uncover novel protein folds that are not yet documented [286].

Fragment Assembly

Fragment assembly techniques construct predictive models using short, contiguous backbone fragments derived from proteins with known structures. This method parallels strategies employed in fragment-based drug discovery, focusing on molecular coherence and structural compatibility. The selection of fragments is typically guided by sequence similarity and the prediction of local structural features, such as secondary structure or backbone torsion angles [161]. In this approach, multiple fragment assemblies are generated and initially assessed using a coarse-grained energy function. The most promising structures are then subjected to a more detailed and computationally demanding fine-grained energy evaluation to determine the optimal prediction model. Historically, fragment assembly has proven to be highly effective, being among the top predictors in several CASP competitions, including CASP9 [213], CASP10 [211], and CASP11 [212].

Machine Learning

Machine learning has a history in protein structure analysis, where it has progressively taken on a more central role from being a supportive component to becoming the backbone of end-to-end predictive models [161, 325]. Early applications were focused on predicting one-dimensional features like backbone torsion angles or secondary structures. Recent advances have seen the evolution of machine learning into

comprehensive approaches capable of predicting entire protein structures. Pioneering methods such as End-to-End differentiable Learning of Protein Structure [10] and the AlphaFold models [144, 325] have proven the potential of machine learning to revolutionise our understanding of protein structures. The success of these machine learning models is mainly dependent on the availability of extensive proteomics data, which provides the necessary training material to refine and validate their predictive capabilities.

3.2 AlphaFold

AlphaFold represents a significant leap forward in protein structure prediction, introduced by Alphabet's DeepMind. The project made its debut in the CASP13 competition [158], where it significantly outperformed all of the competitors in predictive accuracy, using a novel approach based on Convolutional Neural Networks [158]. This introduction of AlphaFold into the scientific community marked the beginning of a new paradigm in computational biology, setting a milestone in the standards for protein structure prediction.

3.2.1 Achievements in the CASP competitions

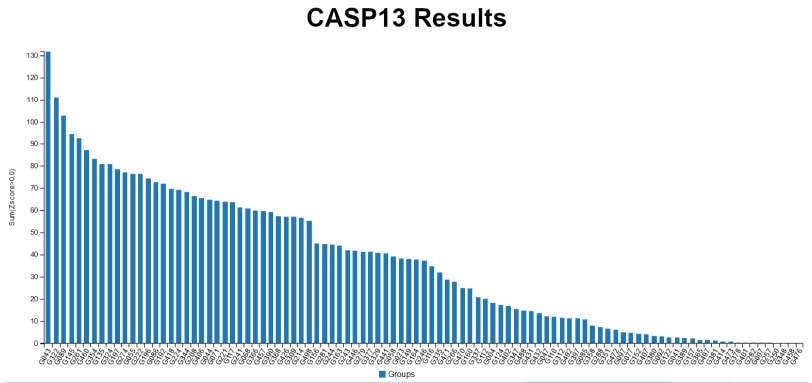
AlphaFold

AlphaFold's entry into the CASP13 competition under the group name A7D marked a cornerstone moment in protein structure prediction. The model demonstrated an unprecedented performance, achieving an anomalous leap in accuracy, which was especially notable in the free modelling (FM) category. It achieved an FM summed z-score of 56.0305, significantly surpassing the nearest competitor, who scored 45.4980. In combining Template-Based Modelling (TBM) and FM categories, AlphaFold achieved a score of 131.6070, compared to 111.0079 by its closest competitor, despite using FM-only techniques [9] (Fig. 3.3).

AlphaFold 2

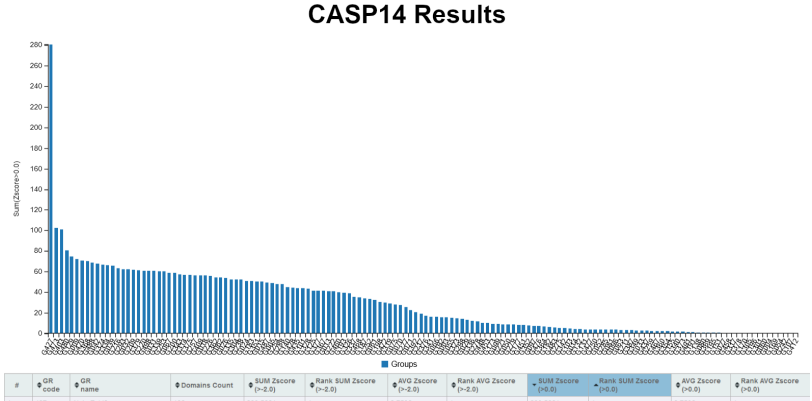
In the subsequent CASP14 competition, AlphaFold 2, also a DeepMind innovation, built upon the successes of its predecessor and set new records [159]. It continued to excel, by comparison with prior years, and significantly outpaced competing entries. The computational methodology employed by AlphaFold 2 allowed it to predict protein structures with a near-experimental accuracy, earning it the description of a "gold standard" in protein folding [6]. AlphaFold 2 dominated the competition with an FM summed z-score of 124.6086, while the closest competitor achieved only 42.7206. Combining FM and TMB/FM categories, its score soared to 280.5661, far surpassing the next-best score of 102.3821 [95] (Fig. 3.4).

Nearly two-thirds of AlphaFold 2's predictions matched the quality of experimental structures, with some even indistinguishable from experimental results [95].



#	GR code	GR name	Domains Count	SUM Zscore (>-2.0)	Rank SUM Zscore (>-2.0)	AVG Zscore (>-2.0)	Rank AVG Zscore (>-2.0)	SUM Zscore (>0.0)	Rank SUM Zscore (>0.0)	AVG Zscore (>0.0)	Rank AVG Zscore (>0.0)
1	043	A7D	107	123.9683	1	1.1586	1	131.6070	1	1.2300	1
2	322	Zhang	107	110.4080	2	1.0319	2	111.0079	2	1.0375	2
3	089	MULTICOM	107	102.2158	3	0.9553	3	102.7383	3	0.9602	3
4	145	QUARK	107	93.8450	4	0.8771	4	94.4160	4	0.8824	4
5	361	Zhang-Server	107	91.6976	5	0.8570	5	92.1633	5	0.8613	5

FIGURE 3.3: The results of the CASP13 competition; the bar graph displays the performance of the competitors, and the table underneath showcases the score of the top performers. The AlphaFold team competed under the name A7D with the group code 043. Source: predictioncenter.org results page [44].



#	GR code	GR name	Domains Count	SUM Zscore (>-2.0)	Rank SUM Zscore (>-2.0)	AVG Zscore (>-2.0)	Rank AVG Zscore (>-2.0)	SUM Zscore (>0.0)	Rank SUM Zscore (>0.0)	AVG Zscore (>0.0)	Rank AVG Zscore (>0.0)
1	427	AlphaFold2	102	280.5661	1	2.7506	1	280.5661	1	2.7506	1
2	473	BAKER	102	101.0821	2	0.9910	2	102.3821	2	1.0037	2
3	403	BAKER-experimental	102	98.2782	3	0.9635	3	100.7841	3	0.9881	3
4	480	FEIG-R2	102	76.9694	4	0.7546	4	80.7039	4	0.7912	4
5	106	Zhang	102	73.7731	5	0.7233	5	74.2589	5	0.7278	5

FIGURE 3.4: The results of the CASP14 competition; the bar graph displays the performance of the competitors, and the table underneath showcases the score of the top performers. The AlphaFold team competed under the name AlphaFold 2 with the group code 427. Source: predictioncenter.org results page [44].

The model demonstrated a median backbone accuracy of 0.96Å root mean square distance and an average error across all predictions of just 1.6Å, equivalent to the width of a single atom.

Outside the confines of the CASP competitions, AlphaFold 2's predictions were validated against the complex structure of ORF3a. Despite the challenging nature of this protein and a lack of similar sequences, the results closely mirrored those determined experimentally later [95], further cementing AlphaFold 2's role as a transformative force in the study of protein structures.

3.2.2 Methodology

The progress achieved by the AlphaFold solutions is underpinned by the methodology employed to predict the protein structure, given its amino acid sequence description. The AlphaFold models employ sophisticated machine-learning approaches that learn the rules of protein folding through statistical means. This section outlines the fundamental components of the AlphaFold approaches, highlighting their fundamental components.

AlphaFold Algorithm

AlphaFold marked a paradigm shift in protein structure prediction through its utilisation of convolutional neural networks (CNNs) and co-evolution-dependent workflow [9]. In order to predict protein structures effectively, AlphaFold leverages evolutionary information through Multiple Sequence Alignment (MSA) statistics. MSA statistics capture clues about the structure of the folded protein; they take advantage of the observation that residues in spatial contact tend to show patterns of correlated mutations [161]. These statistics are used, together with the amino-acid sequence of the protein, to predict the protein's conformation. The architecture of AlphaFold (Fig. 3.5) comprises two main stages: Deep ResNet and Gradient Descent Optimisation.

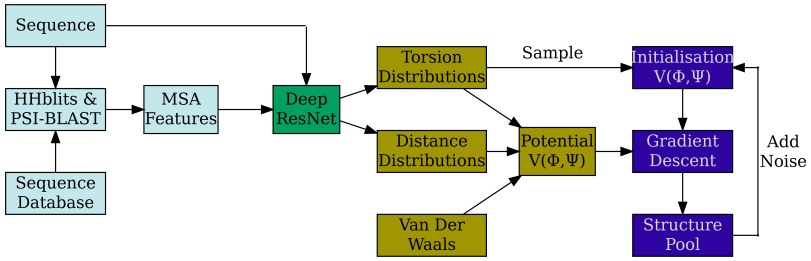


FIGURE 3.5: Component Workflow within the AlphaFold solution architecture.

AlphaFold takes an established approach for depicting interresidue distances (distograms) and employs them in the context of protein structure prediction. They are used as one of the main products of the Deep ResNet together with predicted torsion angles of the backbone Carbon atoms. A distogram represents a 2D histogram of inter-residue distances (Fig. 3.6), effectively converting the 3D structural problem into a more tractable 2D form. This transformation allows the application of sophisticated image processing techniques from computer vision to the domain of protein folding. The histogram format is particularly beneficial as it encapsulates the spatial relationships between amino acid residues within the protein, providing a dense and informative representation of potential folding structures.

Deep ResNet The architecture of AlphaFold ResNet is designed to stabilise the network training and aid in gradient flow; this is essential for the training of very

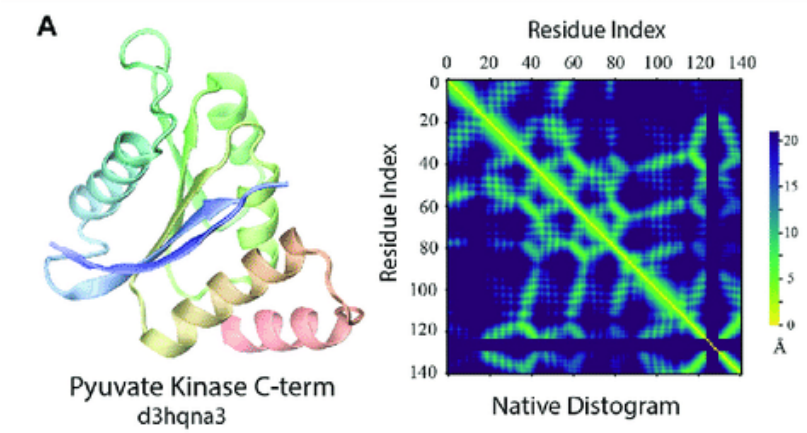


FIGURE 3.6: Visual depiction of the histogram used within AlphaFold. Source: Jumper et al. [144].

deep neural networks. Each block within the network utilises batch normalisation layers and exponential linear units (ELU) to maintain normalisation and introduce non-linearity, respectively. Batch normalisation helps in reducing the internal covariate shift by normalising the layer inputs, helping the network become more stable during training. The residual connections between the blocks help mitigate the vanishing gradient problem by allowing an alternate shortcut path for the gradient to flow through during the backpropagation step. ELUs [298], a version of Rectified Linear Units (ReLUs [3]), apply an exponential function to negative inputs instead of assigning them zero; this further improves the gradient flow compared to the traditional ReLUs.

The core of the network consists of 220 blocks (Fig. 3.7), each of dimension 64x64. These blocks constitute the heart of the ResNet model; within the blocks, the network employs a combination of Batch Normalisation and ELU as its primary processing nodes, repeated three times. Each block can be superficially described as a mini-autoencoder, where the input of the block is processed through a processing node and projected down to a lower, 64-dimensional space. This lower space representation is used within another processing node before going through a 3x3 dilation layer through the last processing node before being projected back to the initial dimension.

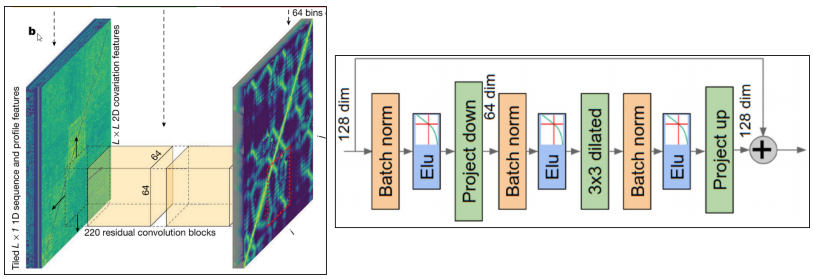


FIGURE 3.7: ResNet model architecture. Source: Jumper et al. [144].

The input to the network comprises Multiple Sequence Alignment (MSA) statistics obtained for the subject sequence combined with the specific sequence for which the structure must be predicted. These are processed through the Deep ResNet to produce a detailed distogram and a prediction of torsion angles for the carbon atoms in the protein's backbone (Fig. 3.5). In order to compute a distogram, the network outputs a probability distribution over 64 equal bins representing a range from 2Å to 22Å for each pair of amino acids. This probability distribution has a dual purpose: first, determining the most probable distance between the two amino acids; second, mapping the network's confidence through the spread of the distribution.

Gradient Descent on protein-specific potential The gradient descent component of the AlphaFodls Architecture is aimed at optimising a protein-specific potential. The computation starts with the establishment of a finely tuned potential field, which is crafted to mirror the protein's intrinsic biophysical properties.

To establish this field, the initial step involves fitting a spline to the negative log probabilities derived from the bins of the distogram. This spline, parametrised by the torsion angles computed by the Deep ResNet, forms the backbone of a differentiable model of protein geometry. This model takes into account the spatial dynamics and constraints of protein structures, ensuring that the predicted shape adheres to plausible physical formations.

The potential field's accuracy is further honed by adjusting for the prior overrepresentation. This is achieved by subtracting a reference distribution from the calculated potential, thus refining the predictive model against biased probability distributions. To mitigate the effects of steric clashes, where atoms are too close to each other, potentially causing physical interference, a smoothing score is integrated. This score incorporates Van der Waals forces, providing a realistic interaction model between the atoms in the protein structure.

Upon establishing the potential field, the torsion angles, which are initially predicted by the Deep ResNet, undergo refinement through a gradient descent process. This step iteratively adjusts the angles to minimise the overall potential energy of the structure, aligning it closer to the conformation with the lowest potential.

AlphaFold 2 Algorithm

One of the ways in which AlphaFold 2 is an evolution from its predecessor is represented by the end-to-end approach of the model. This design paradigm allows the network to fine-tune its predictions more effectively by directly evaluating the actual protein structure, rather than relying on intermediate representations. AlphFold 2 employs a more exploratory approach that can adaptively refine predictions based on direct structural output [143].

Before the prediction process is started, the input data is translated into a deep learning friendly format that the network can manipulate with ease. In this step,

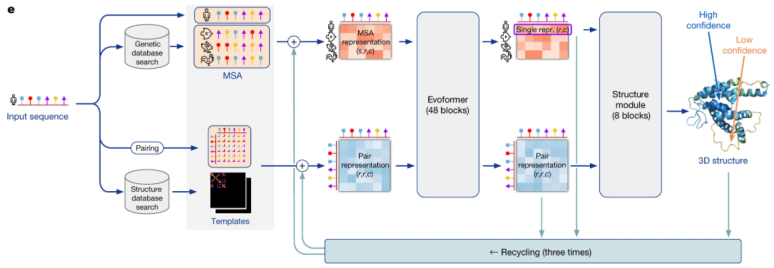


FIGURE 3.8: AlphaFold 2 Model architecture. Source: Yang et al. [325].

the input amino acid sequence is used to query several databases for protein sequences. From these, the solution constructs MSAs that inform the model about evolutionary variations. Additionally, the system searches for existing protein structures that might have similarities with the input sequence, which are referred to as "templates". These templates help construct an initial "pair representation", which models the likelihood of contact between pairs of amino acids in the protein. This information, along with the raw sequence data, is passed through embedding layers that translate this biological data into a format suitable for deep learning models.

AlphaFold 2's architecture (Fig. 3.8) integrates two main components crucial for its performance: the Evoformer that processes the biological data, refining both the embedded inputs, enhancing the accuracy of the model through the relationship between the two embeddings and the Structure module that utilises the data from the Evoformer to construct a detailed three-dimensional model of the protein. The Structure module outputs the Caertesian coordinates for each atom in the protein, including the side chains, effectively creating a comprehensive model of the protein's structure.

The workflow of AlphaFold 2 is iteratively executed a number of times to enhance the precision of the model. Starting with the raw inputs and initial embeddings, the EvoFormer refines the MSA and pair representations through multiple iterations. Following this refinement, the Structure Module leverages the evolved data to build an accurate three-dimensional structural model of the protein in one comprehensive step. The model generated is embedded as MSAs and pair representations, which are then used as input for the EvoFormer until the number of iterations has been fulfilled. This approach ensures that each iteration contributes to a progressively more accurate and detailed protein structure prediction.

Evoformer The architecture of AlphaFold 2's EvoFormer (Fig. 3.9) utilises a unique two-lane architecture consisting of two distinct transformers, each with a specialised function and a clear communication channel between them. This dual structure enables each transformer to focus on different types of data, one on the MSAs and the other on the pair-wise representations, enhancing the model's efficiency and accuracy in handling complex protein simulations.

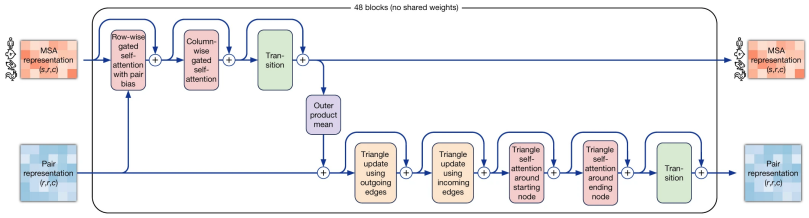


FIGURE 3.9: Evoformer Module Architecture. Source: Yang et al. [325].

The MSA Transformer is designed to compute attention across a vast matrix of protein symbols. To manage the significant computational cost associated with this operation, the attention mechanism is split into two components: row-wise and column-wise. The column-wise attention determines which sequences are more informative while the row-wise attention focuses on determining which pairs of amino acids show more significant relationships and incorporates information from the pair-wise representation.

The Pair Transformer operates in a similar manner to the MSA Transformer with the distinction that it does so on pair-wise representation data, utilising an attention mechanism arranged in terms of triangles of residues. This innovative approach is grounded in the mathematical principle of the triangle inequality, which ensures that the calculated distances between amino acids can be realistically embedded within a three-dimensional space. By enforcing the geometric constraint, the Pair Transformer addresses a common challenge in protein structure prediction, where distance distributions could not previously be effectively translated into three-dimensional configurations.

The components of the EvoFormer work in tandem, iteratively refining the input data through their specialised processes, leading to a more complete and comprehensive representation of the actual structure, informed by hints reinforced by both embedding models.

Structure Module The Structure Module (Fig. 3.10) within AlphaFold 2 introduces a conceptualisation of proteins as a "residue gas" where each amino acid is represented as a triangle corresponding to the three backbone atoms (N, C_α, and C). This representation permits each residue to "float" within a three-dimensional space, simulating the natural fluidity and variability found in protein structures. The module employs affine transformations to manipulate these triangles, effectively moving and rotating each residue to gradually assemble the protein's structure from a dispersed state into a biochemically and structurally consistent form.

Initially, all residues are placed at the origin of the coordinate system, creating a superimposed state of potential configurations. From this starting point, the Structure Module iteratively generates a sequence of affine matrices. These matrices are tasked with spatially displacing and orienting each residue. This iterative transformation process is driven by the network's learning mechanisms, which fine-tune the

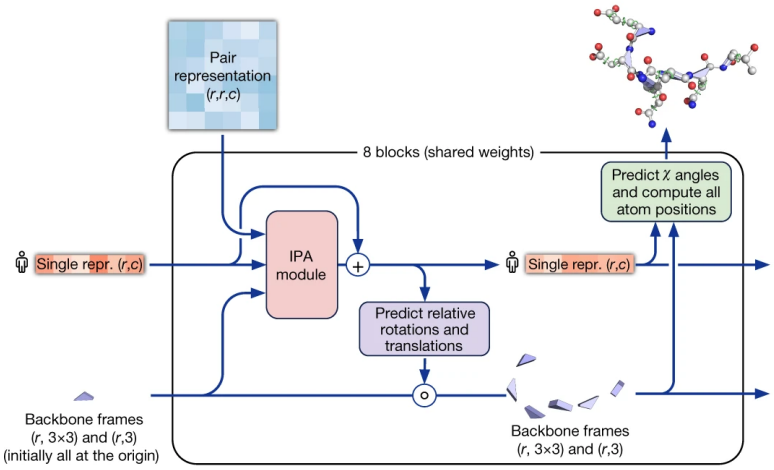


FIGURE 3.10: Structure Module Architecture. Source: Yang et al. [325].

placement and interaction of each amino acid based on evolutionary and biophysical data processed earlier in the EvoFormer.

A novel component integral to this module is the "Invariant Point Attention" (IPA). IPA is specifically designed to handle the three-dimensional nature of protein structures, ensuring that the attention mechanisms remain invariant to translations and rotations. This feature is crucial for maintaining the physical accuracy and consistency of the protein model, as it allows the network to focus on the relational and structural information without being biased by the absolute positions of the residues. By using IPA, AlphaFold 2 remains sensitive to the intrinsic properties of the protein structure while being robust against arbitrary spatial orientations. This invariance property is vital for the accurate prediction and simulation of protein folding dynamics.

The output of the Structure Module is a set of coordinates for each atom in the protein, effectively mapping out the entire three-dimensional structure. This output not only represents the culmination of the predictive process but also embodies the integration of complex biological data through advanced machine-learning techniques.

3.2.3 Improvments of AlphaFold 2 over AlphaFold

AlphaFold 2 represents a major improvement and a complete overhaul of the previous architecture. These changes aim to enhance its predictive accuracy and efficiency in protein structure prediction.

One of the primary enhancements of AlphaFold 2 lies in its utilisation of embeddings that include the full sequences found by MSA along with potential "templates", unlike AlphaFold, which only used MSA statistics. AlphaFold 2's approach provides a richer and more complex understanding of the protein space. By incorporating these detailed embeddings, AlphaFold 2 can capture more nuanced evolutionary and structural information, leading to more accurate predictions [95].

AlphaFold 2 embeds physical and geometric notions directly into its Structure Module rather than relying on a separate search process. This integration significantly restricts the search space, allowing for better performance by excluding physically impossible configurations from the outset. This approach can be viewed as a form of data augmentation; the model inherently understands that any rotation or translation of the data should lead to the same answer. As a result, it requires less data to avoid incorrect models and can learn more efficiently from the available data [143].

Replacing the convolutional neural network with a dual-lane transformer-based architecture marks another critical improvement. The attention mechanism within the transformer allows AlphaFold 2 to dynamically learn information flow, which can incorporate long-range interactions rather than the rigid, local-based information flow of convolutional neural networks. This dynamic learning capability severely enhances the mode's ability to model complex relationships within the protein structure.

AlphaFold 2's design as an end-to-end solution enables the gradient to propagate through the entire architecture. This holistic approach eliminates the disconnect between pair-wise distance predictions and the final 3D structure, ensuring a more cohesive and accurate modelling process. By directly evaluating the cartesian positions of the composing atoms, AlphaFold 2 can fine-tune its predictions more effectively, leading to significantly improved results [143].

These improvements collectively make AlphaFold 2 a groundbreaking tool in protein structure prediction, pushing the boundaries of what was considered achievable in computational biology and setting a new gold standard.

3.3 Computational techniques and technological Innovations

Advancements in computational techniques and technological innovations have been pivotal in the development and success of AlphaFold and AlphaFold 2. These techniques encompass a range of methodologies and innovations that have significantly enhanced the accuracy and efficiency of predicting protein folding.

3.3.1 AlphaFold

AlphaFold's success is underpinned by several key computational techniques that set it apart from previous approaches. Central to its methodology is the use of distograms over distance buckets, which provided a more expressive representation than the traditionally used contact maps. This innovation enabled the application of convolutional neural networks to the problem of protein folding, leveraging their strengths in processing spatial information.

The histogram approach involves predicting the distance between pairs of amino acids in a protein rather than merely identifying whether they are in contact. This

method enhances the expressivity of the model, as it captures more detailed spatial information about the protein structure. By utilising distance buckets, AlphaFold can express both the predicted distances between residues and its confidence in this prediction, offering a nuanced view of the protein's conformation.

To predict these distograms, AlphaFold uses a deep residual convolutional network (Deep ResNet). Residual Deep Networks are particularly effective for such tasks due to their ability to mitigate the vanishing gradient problem, thus allowing the training of very deep networks. The architecture of AlphaFold's ResNet consists of multiple blocks, each incorporating batch normalisation layers and ELU units. These components help stabilise the training process and improve the gradient flow, reinforcing the model's robustness.

Another representative component of AlphaFold's architecture is the differentiable module for spline fitting over the predicted histogram. This module takes the histogram output and fits a smooth spline, which represents the protein-specific potential surface. This differentiable approach allows for the integration of gradient descent optimisation directly on the potential field, refining the torsion angles and ensuring that the predicted structure adheres to biophysical constraints.

These computational techniques collectively contribute to AlphaFold's anomalous leap in accuracy for predicting the protein's fold and demonstrating the potential of AI-driven approaches in addressing complex biological challenges.

3.3.2 AlphaFold 2

AlphaFold 2 represents a breakthrough advancement in protein structure prediction. By incorporating a more sophisticated representational mechanism through its novel architecture, AlphaFold 2 can predict protein structures with unprecedented accuracy. Two of the most notable key innovations of its architecture include the dual-lane transformer-inspired module, known as EvoFormer, and the Invariant Point Attention mechanism utilised within the structure Module Transformer.

The EvoFormer processes biological data in a unique two-lane architecture, each lane dedicated to handling different types of data, one for MSAs and the other for pair-wise representations. This dual structure enables the EvoFormer to efficiently manage complex protein simulations by computing attention across a complex set of protein sequences and their interactions. This design interweaves information pieces from both embedding models, constructing a more accurate representation of the protein structure.

The Structure module employs the tailored style of the attention mechanism titled Invariant Point Attention, specifically designed to manage the three-dimensional nature of proteins. The IPA ensures that the attention mechanisms remain invariant to translations and rotations, maintaining the physical consistency of the predictions.

Another technique used by AlphaFold 2 to enhance its training is self-distillation. The model is initially trained on the 170,000 protein structures available in the PDB. This intermediary model is used to make structural predictions for an additional

350,000 sequences. High-confidence predictions from these sequences are selected and combined with the original PDB data, creating a new enriched dataset. The model is then retrained from scratch using this augmented dataset, significantly improving its performance by incorporating a broader range of structural information and nuances from both known and newly predicted protein structures.

These are some of the key computational techniques and innovations that collectively contribute to making AlphaFold 2 a groundbreaking tool in computational biology, setting a new standard for protein structure prediction.

3.4 Potential applications in drug discovery

The advent of AlphaFold 2 has ushered in a new era for drug discovery, potentially revolutionising the various steps of the Pre-Clinical stage and the computational approaches in the pharmaceutical industry.

AlphaFold 2's prediction will become integrated into experimental workflows for structure prediction. Traditionally, determining the structure of proteins using methods like X-ray crystallography and NMR spectroscopy is time-consuming and resource-intensive. By providing high-accuracy initial models, AlphaFold can significantly expedite this process. For instance, AlphaFold 2's predicted structures can serve as starting points for fitting the electron density maps obtained from X-ray crystallography experiments. This integration could streamline the interpretation of complex proteins, reducing the time required to achieve a definitive answer.

AlphaFold 2 will be employed as a vital step in virtual screening within the drug discovery pipeline. By providing accurate protein structures, AlphaFold enhances the reliability of these virtual screening processes, enabling the identification of promising drug candidates with high precision. This capability is particularly crucial for structure-based drug discovery, where the knowledge of a target protein's structure guides the design and optimisation of therapeutic molecules.

Furthermore, the data generated by AlphaFold 2, if not the model itself, holds immense potential for future developments in reverse drug design, where, given the shape of a binding pocket or a specific interaction site, researchers design a protein that would fold in a manner compatible with this site. The comprehensive structural data provided by AlphaFold 2 can lay the groundwork for innovative methodologies, fostering the development of new therapeutics.

AlphaFold 2 is posed to become an indispensable tool in drug discovery. Its integration into experimental workflows for structure prediction, application in virtual screening processes, and contribution to structure-based drug discovery highlights a few of the high-impact areas in drug discovery that are going to be transformed by the predictions of AlphaFold 2, setting the stage for groundbreaking advancements in the pharmaceutical industry.

3.5 Technical Challenges

The birth of AlphaFold and its subsequent versions have reshaped the landscape of protein structure prediction, propelling the capabilities of computational biology to unprecedented heights. However, with groundbreaking advancements come unique sets of challenges that underscore the complex nature of protein folding and machine learning.

3.5.1 AlphaFold Limitations

Despite its significant success, AlphaFold's approach incorporates methodologies that carry intrinsic limitations which can affect the accuracy and generalisability of its predictions. One major limitation arises from the use of CNNs, which, although excellent for pattern recognition in images, are less suited for sequence-based predictions such as protein folding. CNNs are inherently designed to be translation-invariant; they detect features regardless of their spatial location within the input data. This property, while advantageous for tasks like image recognition, leads to potential losses of crucial positional information in protein structures where the specific locations of amino acids are vital for accurate modelling.

Another limitation of CNNs is that they primarily capture local dependencies within their receptive fields, which makes them ideal for identifying localised image features but less effective for protein structures where long-range interactions significantly influence folding. This limitation can result in the network overlooking critical interactions between distant amino acids, potentially leading to inaccuracies in the predicted structures.

AlphaFold's design choice of using histogram encoding to extract conformational information from the available biological data represents another challenge. While distograms provide ample representational power, the actual space of protein folds is much smaller compared to the vast array of possible histogram configurations. This discrepancy can lead the network to predict distograms that are not representative of viable protein structures, thus expanding the search space unnecessarily and reducing the overall accuracy and efficiency of the model.

The process of converting the distograms into a smooth potential field through spline fitting introduced another informational bottleneck. This step simplifies the data at the cost of potentially discarding subtle yet crucial details and confidences captured by the network. Such information loss can be detrimental to modelling the intricate dynamics of protein folding.

3.5.2 AlphaFold 2 Limitations

Despite its remarkable achievements, AlphaFold 2 is not without limitations. One significant challenge arises from the inherent nature of deep neural networks, such as transformers, which AlphaFold 2 employs. While these models excel in predicting

protein structures, they do not inherently provide insights into the underlying mechanisms of protein folding. This limitation leaves the model open to Smart Aleck type of vulnerabilities, where the model may generate plausible but incorrect or misleading predictions due to the lack of understanding of the underlying physicochemical processes involved.

Another limitation concerns the reliance of AlphaFold 2 on MSA. MSAs are dependent on existing data, constraining the model from interpolating between known protein structures. This dependency can hinder the model's ability to predict novel or underrepresented protein configurations. To address this, a potential solution would involve pre-training the model on molecular dynamics simulations to learn the physicochemical rules governing protein structures, making use of the foundational training approaches used in LLMs.

The training data sourced from the PDB also imposes some limitations. The structures in the PDB may not always represent the natural state of proteins; they often reflect proteins in specific conditions, such as in complexes with other molecules. This specialisation can skew the model's learning, focusing on protein conformations that are not universally applicable outside these specialised states. Such conditions are usually noted during experimental resolution but may not be adequately accounted for by the neural network, potentially limiting its ability to generalise across different biological contexts.

3.6 Future Directions

Observing other fields that successfully employ the Transformer architecture, such as LLMs, indicates that the performance of these systems scales proportionally with increases in model size and training data volume. Given the limited availability of experimental protein structure data, future improvements may necessitate an innovative approach to training, such as pretraining on molecular dynamics simulations to enrich the data and incorporate more refined physicochemical protein properties.

The field is likely to see the development of foundational models for protein folding, comprehensive, robust models trained on extensive datasets that can generalise across a range of tasks, from predicting basic structures to mapping complex interactions. These models would standardise high-level predictions in the field, similar to the foundational models in natural language processing.

Focus may also shift towards previously challenging areas, like membrane proteins, which are difficult due to their dynamic natures and the environments in which they operate. Future AI models will grow to cover these more challenging areas for prediction.

Despite all the advancements, a significant gap remains in understanding the underlying mechanisms for protein folding. Addressing this gap is crucial not only for improving model reliability but also for enhancing our fundamental understanding

of biological processes. Efforts in AI explainability could play a pivotal role, providing insights into the decision-making process, highlighting the relevant portions of the data that the network uses in processing its information and possibly paving the way for new breakthroughs in drug development and biotechnology.

3.7 Conclusion

Throughout this chapter, we have explored the groundbreaking achievements of AlphaFold and AlphaFold 2 in protein structure prediction, highlighting their functionality and impact on the field of computational biology. These technologies have not only redefined the benchmarks for predictive accuracy but have also significantly broadened our capacity to understand and manipulate protein structures with unprecedented precision.

AlphaFold revolutionised the approach to protein structure prediction by integrating convolutional neural networks and introducing a novel gradient descent-based module for transforming distograms into protein structure; this approach has propelled AlphaFold to the top of the CASP XII competition. Its successor, AlphaFold 2, further advanced these capabilities by employing sophisticated architectural innovations like dual-lane transformers and invariant point attention, achieving near-experimental accuracy and setting a new gold standard for the field.

Chapter 4

Molecular Recognition, Puzzle Solver

Chapter 4 expands on the work of Chapter 3 and builds upon the research published in "Silver Surfer: Navigating the Parametric Protein Space with Genetic Algorithms" [194], shifting the focus from conformation determination to molecular recognition through the exploration of jigsaw puzzle solver, a simplified analogous problem. The chapter presents a realistic jigsaw puzzle generator, introduces a novel AI-based solution, elaborates on the knowledge gained from building the model, and concludes with possible future directions in molecular recognition design.

4.1 Introduction

The challenge of assembling a jigsaw puzzle, an engaging and familiar task, known to most people since childhood, mirrors the intricacies and complexities of molecular recognition in protein design. Both domains require the identification and matching of uniquely shaped pieces (puzzle pieces or molecular structures) to form a functional assembly. This analogy simplifies the abstract concept of molecular interactions and frames a complex scientific problem in a more tangible, relatable context. In the realm of computational biology, the techniques developed for automated jigsaw puzzle solving provide a unique perspective on molecular recognition, a critical process in the drug design pipeline. The objective of this chapter is to tackle the simplified problem in order to extract knowledge and inform the development of the more complex task of molecular interactions.

4.1.1 Connection with Protein Design

The second step in the drug development pipeline is hit target identification. This stage is critical as it lays the foundational groundwork for the drug design process. Identifying qualitative hits involves determining compounds that exhibit effective drug potential through favourable interaction with specific biological targets. The accuracy and efficiency of this process significantly influence all the subsequent stages of drug development, making it a cornerstone of pharmaceutical research.

The problem-solving process in both jigsaw puzzles and drug design involves determining which component (or piece) interacts (or matches) another within an assembly. This analogy describes the methodologies used in both fields. In jigsaw puzzles, each piece must connect with its correct neighbour based on shape and pictorial in order to form a complete picture. Similarly, in drug design, the right molecular or chemical compounds must be identified that can 'fit' or bind to specific molecular targets, akin to finding the right piece of the puzzle. This matching process is fundamental to both puzzle-solving and the hit identification stage of drug design.

The lock and key mechanism, though slightly outdated, provides a fundamental description of how molecular recognition and puzzle-solving work. It describes how two molecules must have complementary shapes and properties to bind effectively, similar to how puzzle pieces fit together. Although this model has been refined over the years, its basic premise remains valid and is supported by modern theories. These theories debate not whether shape complementarity is necessary but rather how these conformations are achieved in dynamic biological environments.

Machine Learning, and particularly deep neural networks, have shown significant promise in learning these complex associations. By training on vast datasets that include numerous examples of molecular interactions, neural networks can learn to predict which molecules will interact based on their structural and chemical properties. This capability can be likened to solving highly complex, multidimensional jigsaw puzzles where each piece represents a potential pharmaceutical agent.

4.1.2 Motivation on tackling the problem

The motivation behind addressing the jigsaw puzzle challenge in the context of molecular recognition stems from the analogous challenges both domains present. Jigsaw puzzles, much like the molecular puzzles faced in protein design, involve the assembly of unique pieces into a correct configuration. This process is a computationally complex one, classified as NP-complete [69]. By approaching this well-defined, simpler two-dimensional problem, we aim to distil effective strategies and insights that can be translated into the more complex, multi-dimensional world of protein interactions.

The approach of attempting to solve a simplified version of the problem first, adheres to the principle of "learning to walk before learning to run'. It is through mastering the techniques for solving simpler, controlled puzzles that insights into the complexities of the more chaotic and complex field of molecular recognition can be gathered. The step-by-step progression from jigsaw puzzles to 3D puzzles, ending in protein docking, is designed to incrementally build the capabilities and understanding of the problem, ensuring that each challenge tackled while dealing with simplified challenges can illuminate paths forward in the intricate landscape of drug development.

Tackling the jigsaw puzzle challenge serves a dual purpose: it advances our computational techniques and engages a broad audience with its familiarity and inherent challenge. The engaging nature of puzzles provides an intuitive gateway for explaining the intricate process of hit identification in drug discovery.

The exploration of machine learning methods for solving jigsaw puzzles offers a compelling preliminary model to test, refine, and understand strategies that could be pivotal in advancing molecular recognition.

4.2 Theoretical Foundations: From Jigsaw Puzzles to Molecular Recognition

The analogy between assembling jigsaw puzzles and understanding molecular recognition in protein design is profound yet insightful. The simplicity of the jigsaw puzzle serves as an ideal model to unravel the complexities of molecular interactions, particularly in the context of hit identification for drug design.

Jigsaw puzzles require the accurate segmentation and correct assembly of interlocking pieces, each with a unique shape and design. Similarly, in molecular recognition, researchers must extract meaningful information from protein sequences, a process similar to discerning puzzle pieces from a nondescript background. Each protein or molecule, like a puzzle piece, must be evaluated not only for its shape but also for its ability to fit with the target protein in a process highly reminiscent of solving a jigsaw puzzle.

The matching step in jigsaw puzzles, where each piece is connected to form a coherent image, mirrors the molecular recognition process, in which proteins interact through their complementary shapes and chemical properties. To illustrate this concept further, consider a protein's 3D (Fig. 4.1 A.) structure and its surface shape and properties (Fig. 4.1 B.), such as electric charge, which can be visualised as complex three-dimensional jigsaw puzzle (Fig. 4.1 C), where the surface properties provide hints similar to the chromatic clues found in each puzzle piece.

Just as in jigsaw puzzles, where the vast number of possible piece combinations far exceeds the number of correct configurations, the biological world presents an even greater challenge. The number of potential interactions between molecules is enormous, yet only a fraction of these interactions are active, viable, biologically relevant and therapeutically useful. Navigating this immense complexity requires sophisticated tools and approaches, many of which can be informed by the strategies involved in developing an automated solution for jigsaw puzzle solving.

4.2.1 Description of the Jigsaw Puzzle Challenge

The jigsaw puzzle, a quintessential part of many childhoods, has been a source of entertainment for over three centuries. First recorded in 1760, the jigsaw puzzle was invented by a London map maker named John Spilsbury [277]. Originally designed

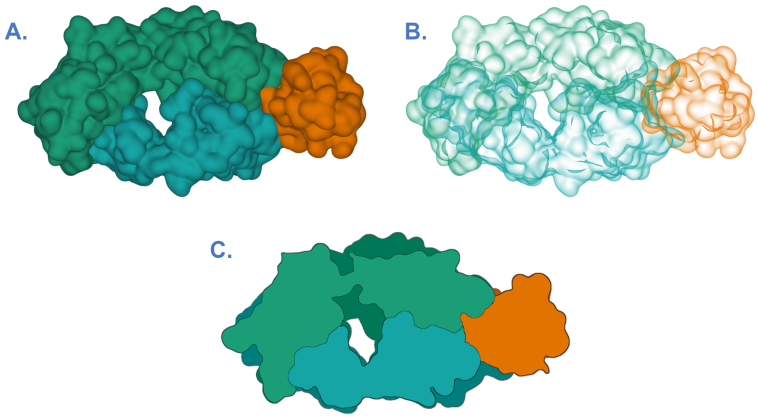


FIGURE 4.1: Images of the 2JEL [237] protein, visualised with the help of MOL [273] on the RCSB website [230] reveal the similarity between the jigsaw puzzle problem and Protein-Protein interaction. A. - the Gaussian surface representation. B. - the x-ray representation. C. - Gaussian surface intersected with a clipping plane.

as an educational tool to teach geography, these puzzles were done by cutting a map printed on a wooden board into various interlocking pieces. Beyond their role in leisure and education, jigsaw puzzles have been recognised as a marker of meta-cognitive development [77]. The ability to solve pictorial jigsaw puzzles reflects advanced problem-solving skills and spatial reasoning.

The formal definition of a classical jigsaw puzzle defines it as a tiling puzzle consisting of interlocking, irregularly shaped pieces. Each piece typically has a portion of a pictorial printed on it; when assembled correctly, they reveal a complete, coherent pictorial as depicted in Fig. 4.4 subpoint D. The classical version of a jigsaw puzzle often comprises 18 types of pieces, depicted in Fig. 4.2, that interlock in specific configurations.

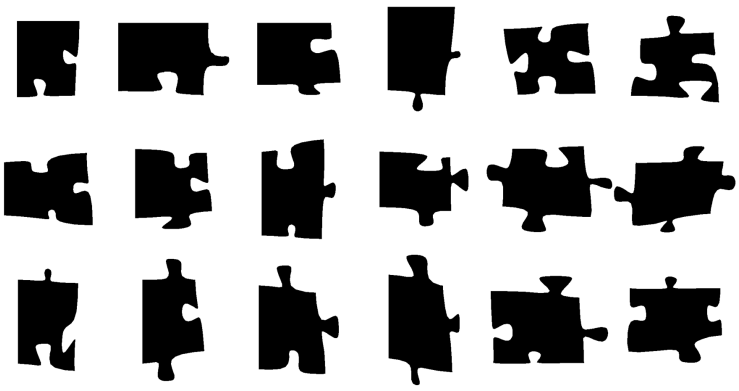


FIGURE 4.2: Representatives of each of the 18 classical types of jigsaw puzzle pieces.

The specific version of the problem tackled revolves around the most common rendition of the jigsaw puzzle, which is to iteratively connect matching piece pairs

in order to correctly reconstruct the pictorial of the puzzle. The pieces are presented in a scrambled configuration on a uniform textured background, all visible, non-overlapping and facing up, randomly displaced translational and rotationally. The puzzle tackled is pictorial, connected, and regular, with unknown dimensions. Each puzzle piece is distinct, showing slight variations from one of the 18 standard types. Internal pieces have four connecting neighbours, edge pieces have three, and corner pieces have two. In the solved state, these pieces are arranged in a grid-like layout, where internal connections form quad radial junctions resembling a "+" sign and edge connections form triradial junctions similar to a "T". The ultimate objective of this challenge is to find and attach all the puzzle pieces to reveal the hidden image or message.

4.2.2 Historical Approaches

The first attempt at an automated jigsaw puzzle solver occurred in 1964, spearheading the use of computational methods in solving complex pattern analysis problems, supported by the Air Force Office for Scientific Research [260]. A next major milestone was in 1994 when D.A Kosiba et al. introduced "An automatic jigsaw puzzle solver" that utilised both shape features and colour information for matching puzzle pieces [153]. This approach improved accuracy by considering multiple visual attributes. The evolution continued with M Sağiroğlu et al. [259], which shifted the focus entirely towards colour and texture, employing in-painting synthesis to predict and match textures outside a piece's boundary, demonstrating a new strategy in tackling the jigsaw puzzle challenge.

As computational capabilities expanded, SOTA solutions began employing various dissimilarity/compatibility functions to enhance the precision of piece matching. These methods, including sum-of-squared differences for patches along the piece boundary [55], gradient intensity changes for patches along the piece edge [100] and checking lines of pixels at the edge [110, 226, 235, 277]. The most common approach is to solve the puzzle in a greedy fashion, selecting pairs of pieces based on the highest compatibility scores.

One notable non-greedy approach is presented by Sholomon et al., who propose a technique, a genetic algorithm approach to solving jigsaw puzzles [277]. In their approach, the dissimilarity/compatibility function is summed up over all the neighbours of every piece, and the resulting score is used as the fitness score for the candidate. The proposed approach is capable of solving puzzles with impressive dimensions but requires that both piece orientation and the size of the puzzle be known; this significantly reduces the scope of the jigsaw puzzle challenge, essentially transforming it into finding the correct permutation. Despite their imposed prerequisite, this method demonstrates the capability of Machine Learning approaches to tackle the puzzle-solving problem.

The application of machine learning in solving puzzles was further explored by Paumard et al. through the development of a deep learning-based model titled

Deepzzle [229]. This model tackled a jigsaw-like problem using deep neural networks and focused on scenarios where edge information might be lost, such as in archaeology, further demonstrating the potential for deep learning techniques to be adapted to various real-world applications.

Despite the progress and technological strides, most solutions assume that puzzle pieces are already correctly segmented from their background; a non-trivial task that remains a significant challenge in both puzzle solving and molecular recognition, where the correct identification and segmentation of active sequences are crucial for accurate drug design.

4.2.3 Training Data Requirements

Deep neural networks require extensive, well-annotated datasets to effectively learn complex patterns, a challenge magnified in specialised applications like jigsaw puzzle solving, where annotated data is scarce or non-existent. Compiling and annotating such a vast dataset manually would be prohibitively costly and time-consuming.

To address this, a virtual puzzle generator capable of synthesising realistic jigsaw puzzles was developed and made publicly available. This tool automates the generation of diverse puzzle pictures, ensuring a steady supply of high-quality data for training deep learning models. By mimicking real-life puzzles in various configurations and complexities, it provides a controlled environment for both training and validating our models.

The generated images are used extensively in the training of the model, which can be, at a later stage, fine-tuned on a much more limited set of real-world data to improve its performance in specific applications. Training on synthetic data bypasses the need for manual data collection and virtually removes both the problems of overfitting and the size limitations imposed by limited training corpora.

Generating synthetic data to overcome challenges associated with data scarcity and annotation in machine learning can be a very effective solution. The performance of this approach is highly dependent on the distribution similarity between the real encountered data and the synthetically generated data.

4.2.4 Bézier Curves

Bezier curves represent one of the foundational tools employed in the proposed jigsaw puzzle generator and play a role in the design of interlocking pieces. Bźier Curves are parametric curves, as illustrated by figure Fig. 4.3, that are defined by a mathematical formula (4.1) which utilise a set of control points to determine the curve's shape.

The general formula for a Bezier curve with n control points is expressed as:

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^i P_i \tag{4.1}$$

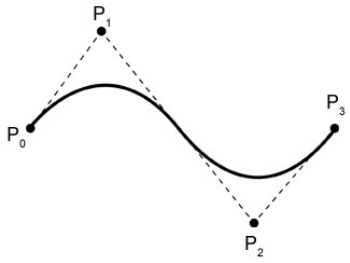


FIGURE 4.3: Example of a bezier curve with 4 control points.

In the formula for Bézier Curves (4.1), t is a parameter that ranges from 0 to 1, representing the interpolation distance from the start to the end of the curve. The curve itself is shaped by n control points, $P_0, P_1, \ldots, P_n$.

The decision to employ Bézier curves in the design of puzzle pieces stems from their capacity to model various complex shapes, which is crucial for the unique and intricate design of jigsaw puzzle joints. The Significant properties of Bézier curves that motivate their use in the puzzle generator include:

Endpoint Control. Bézier curves start and end at the first and last control points. This feature ensures precise alignment at the joints, the intersections between pieces, and at the edges of the puzzle, enabling complex designs while guaranteeing that the connections meet accurately at designated points. This precise control is vital for the proper alignment of the cutting template.

Subdivision. Bezier curves can be segmented into any number of smaller sub-curves, each retaining the properties of a Bezier curve. This allows for a modular approach to designing puzzle joints, where the design can focus on individual joints rather than having to design the entire row or column simultaneously. This modularity simplifies the design process and allows for varied puzzle piece shapes.

Convex Hull. The curve does not extend outside the convex envelope formed by its control points. This property provides precise control over the area where cut-outs can be made, ensuring that the design does not exceed the boundaries of the specified segment of the puzzle piece. Such precision ensures the integrity and connectivity of the puzzle pieces.

By leveraging the mathematical properties of the Bézier curves, the puzzle generator can create intricate and varied puzzle shapes.

4.3 Proposed Solution

This section outlines a comprehensive solution that addresses the challenges posed by the jigsaw puzzle problem. The proposed solution utilises STOA machine-learning approaches for matching puzzle pieces together with a novel jigsaw puzzle generator employed in the training of the model.

4.3.1 Jigsaw Puzzle Generator

The Jigsaw Puzzle Generator is designed to create a highly accurate simulation of scrambled jigsaw puzzles. This generator mimics the real-world process of puzzle fabrication, thus providing a robust tool for generating the training and validation sets for machine learning models used in the automatic collection of jigsaw puzzles. The generator's design ensures that the puzzles it creates closely imitate the appearance of actual scrambled puzzles.

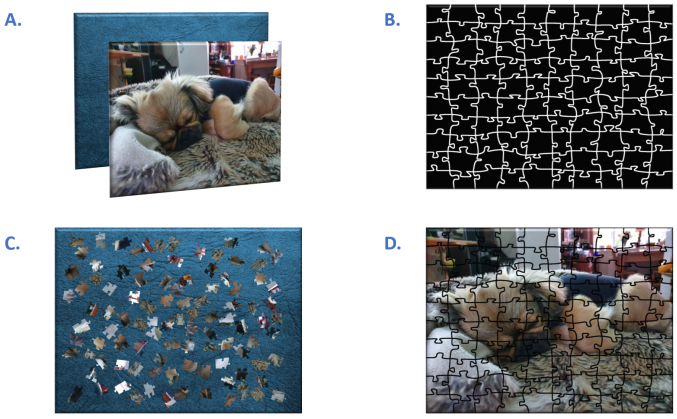


FIGURE 4.4: A. - Puzzle Pictorial and its selected background texture, B. - Cutout template generated with the help of the puzzle generator, C. - jigsaw puzzle scrambled over the background texture, D. - Correctly assembled jigsaw puzzle.

The generation process begins with the selection of an image to serve as the pictorial of the puzzle, depicted in Fig. 4.4 A. This choice dictates the subsequent dimensions of the puzzle, establishing the scale at which the puzzle will be constructed.

With the dimensions established, a new cut-out template matching the puzzle's size is generated (Fig. 4.4 B.). The template divides the image into the desired number of rows and columns. Each square is designed to form a peg/socket joint at its edges, adhering to the traditional grid layout of puzzles. This structure is critical for the interlocking nature of puzzle pieces.

Each internal edge joint is defined with the help of Bèzier curves. These curves facilitate the creation of intricate connections that are characteristic of jigsaw puzzles, allowing for varied and unique puzzle piece shapes. The horizontal cut-out templates are defined first, with a joint separately designed for each column. The peg/socket joint is defined using a random number of control points positioned strategically in order to generate the characteristic shape. In order to increase the influence of a control point while keeping it in the area designated for connections so that the security provided by the convex-hull property is maintained, a strategy of placing multiple points at the same coordinates is employed. This significantly increases the influence, pulling the line towards the desired direction while maintaining the guarantee that the line will not go out of bounds of the joint designated

area. The vertical joints are designed using a similar approach, with the additional check to avoid overlaps or tangential touches, ensuring that the puzzle piece remains one connected component.

Once the cutout template is generated, a background image is selected to enhance the similarity with real puzzle pictures. The background image is selected to be sufficiently large to accommodate all pieces in a non-overlapping layout; empirical experimentation has shown that a background image twice the size of the pictorial image works optimally. A patch of the required size is cropped from the background image, a random offset, and manipulated (flipped and mirrored) to increase the diversity of the available backgrounds, making it more difficult for the model to overfit over the backgrounds.

Once cut according to the cut-out template, the pictorial pieces are randomly placed against the background. Each piece undergoes a random roto-translational transformation to simulate the random scattering in an actual puzzle. A final blurring step is applied to complete the image (Fig. 4.4 C). This is done in order to soften the transition between pieces, mimicking the aesthetic of real puzzle pieces.

The generator is publicly available for research purposes and can be accessed through the dedicated GitHub (`https://github.com/Bogst/JigsawPuzzleGenerator`) repository. This tool is intended to foster further research and development in the field of computational jigsaw puzzle solving.

4.3.2 AI Solver

To tackle the Jigsaw puzzle challenge, a solution that incorporates the use of a deep convolutional network architecture is proposed. One of the main reasons that motivate the choice of this approach is the necessity to segment the puzzle pieces from their backgrounds accurately, a task for which CNNs are producing SOTA results [203].

The problem of detecting the puzzle piece on a background image shares superficial similarities with traditional object segmentation tasks. However, the key distinction lies in the nature of the puzzle pieces themselves, which can contain varied graphical elements that the network needs to disregard to segment the pieces accurately. The network needs to learn the 18 types of puzzle pieces to be able to accurately distinguish between puzzle pieces and backgrounds. This difference is sufficient to necessitate training the network from scratch rather than relying on fine-tuning an existing model.

In the initial training phase, the network undergoes pre-training on an autoencoder task, which segments the puzzle pieces from their background. This step is crucial as it helps the model to understand the edges and shapes of the pieces, which are vital for accurate detection and matching. This task also reinforces an internal representation of the different kinds of puzzle pieces.

After pre-training, the decoder component of the network is replaced with a fully connected network architecture that culminates in four output heads. Each of these

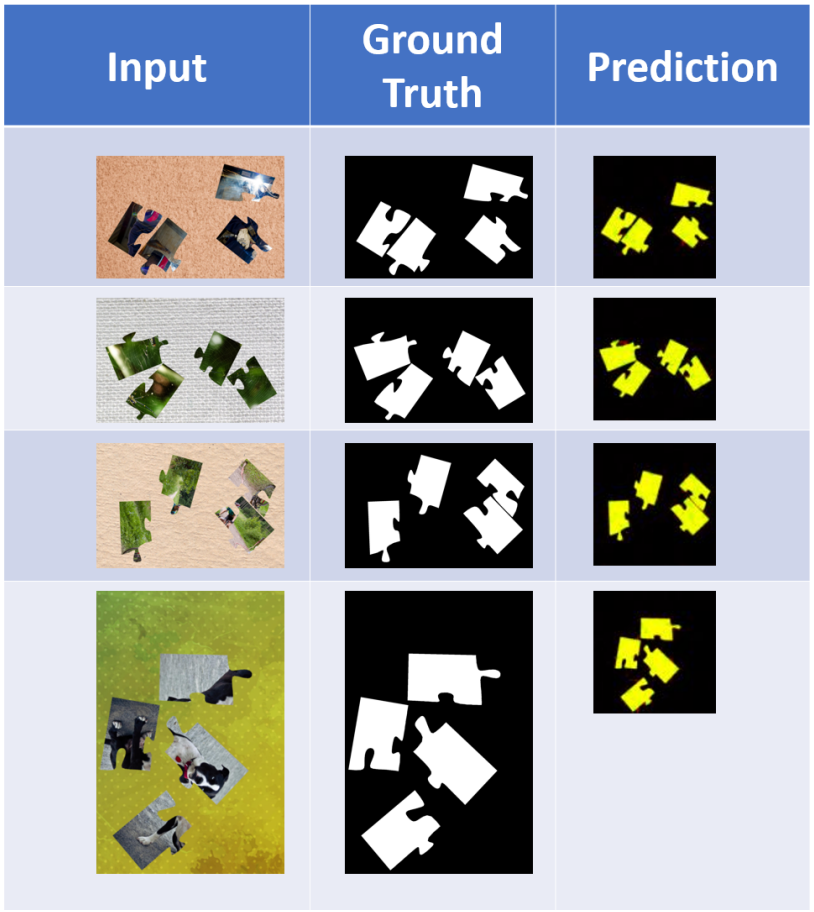


FIGURE 4.5: Results of the encoder-decoder training; Green (false negatives), Red (false positives), Yellow (true positives).

heads is tasked with predicting the exact coordinates of correct socket/peg connections for a four-piece jigsaw puzzle configuration. The network processes the embeddings from the encoder component in order to predict the correct coordinates of matching connections. A fine-tuning stage follows once a plateau in performance is reached, during which the entire network is updated to enhance accuracy further.

This end-to-end approach is preferred over a segmented method involving separate components for piece segmentation and matching as it allows a richer flow of information through the model, enabling it to leverage complex details about the puzzle pieces and their arrangements. As part of the segmentation stage, the network must learn and encode the different classes of pieces; this information would have to be learnt again by the second component if an approach with two different components is used. This leads not only to an information bottleneck but also to the duplication of the learned contents. The continuous flow of information of an end-to-end approach enables an easier understanding and utilisation of the intricate details of the pieces, a crucial element for accurate puzzle assembly.

The architecture of the proposed model begins with inputs of resized images of dimensions 98x98x3, processed through four convolutional layers. The first two layers employ filters of dimension 5, followed by two layers using filters of dimension

3. The outputs from these layers are flattened and processed through several fully connected layers with decreasing sizes, culminating in the four output heads.

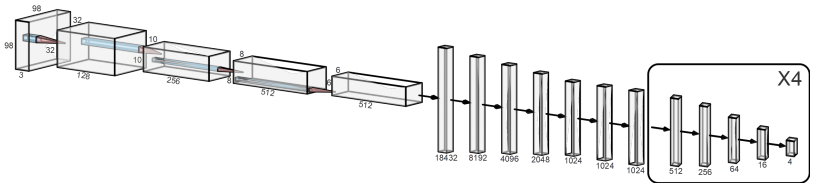


FIGURE 4.6: Puzzle solver architecture.

To preserve the precise localisation of connection points, which is critical for assembling the puzzle correctly, the utilisation of pooling layers was omitted. Despite their widespread use together with convolutional layers in order to further decrease the dimensionality of the input data, they also introduce uncertainty with respect to the exact coordinates, which would negatively impact the performance of the model.

The network's training uses the Euclidean pixel-wise distance from the true centre of peg/socket connections as the loss function, optimised using the Adam optimiser. The model reached a validation loss of 423.859px (4.7), demonstrating the capacity to predict not only matching pieces but also what part of the piece is the active part in the match, a feature that is highly desirable in the problem of molecular recognition.

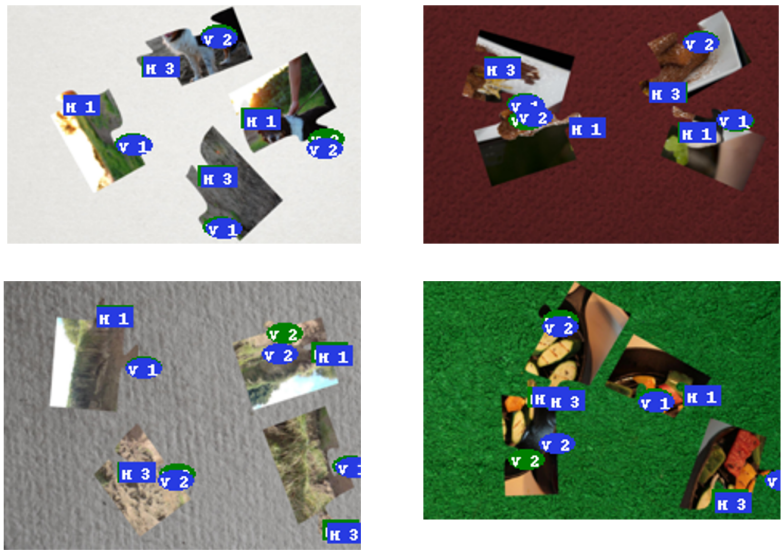


FIGURE 4.7: Examples of the solvers' connection predictions (blue) over true connections (green).

Development and training were realised with the help of the TensorFlow platform on a system equipped with an Intel I7-10750h CPU, 16GB of RAM and an NVIDIA RTX 2060 GPU.

The initial training setup linked the generator directly into the training of the network, producing on-demand samples. The training was later updated to train off of an existing dataset due to the high computational demand of real-time data

generation and limited improvement. The dataset comprised 100,000 pictures of scrambled jigsaw puzzles for training and a set of 8,780 pictures utilised for validation. The images employed are a subset of the Open Images Dataset, while the background pictures have been manually scraped from the internet.

4.4 Lessons from the AI-Solver

Developing the AI-Solver has provided a number of insights directly applicable to the more complex domain of protein docking.

4.4.1 Knoweledge Gained

The success of the relatively small Deep Learning network in resolving the jigsaw puzzle problem showcases a promising solution for approaching the inherently more complex task of molecular recognition. The puzzle solver, while conceptually simpler, mirrors some of the intricacies of protein docking, where the correct matching of molecular structures and surface properties is key. This project demonstrated that even a limited model can grasp and execute complex and abstract meta-pattern recognition tasks. This success indicates a strong potential for a large deep-learning network to model the nuanced interactions of molecular recognition.

The development of the solver reinforced that both the quantity and quality of the training data are paramount for achieving high performance. High-quality, well-annotated data is an unavoidable prerequisite for training the model to recognise and accurately pinpoint the location of matching puzzle pieces. Similarly, in the domain of molecular recognition, the availability of comprehensive and qualitative molecular data sets will be critical in training models to predict interactions accurately.

For the training of the AI-Solver, the ability to generate synthetic data proved invaluable in overcoming the limitations associated with scarce training data. This approach facilitated a continuous supply of diverse training samples and ensured that the model could generalise well over real-world data by exposing it to a wide range of samples. This strategy might be necessary for molecular recognition, especially in areas where experimental data may be limited or very costly to obtain.

Utilising unstructured data in a pertaining phase, such as through tasks of inpainting, proved a highly effective way to extract knowledge out of unlabelled data. This approach allowed the model to learn useful features from a broader data set, enhancing its ability to focus on relevant features. This strategy is particularly valuable in fields like molecular recognition, where annotated data might be limited. The network could learn general protein mechanics from unlabeled sequence data and then refine and select the relevant mechanics based on the limited available data.

One of the challenges encountered was the model's tendency to get stuck in local minima, predicting an average of all samples when not guided adequately through

the training schedule or loss function. This underlines the importance of defining the problem clearly and ensuring the model has sufficient representation power to escape these traps. Crafting a loss function that encapsulates the complexity of the task and effectively guides the optimisation process is crucial. This lesson is directly translatable to the problem of molecular recognition, where defining the right loss function will determine the success of predicting accurate molecular interactions.

The project also highlighted the importance of effective versioning and logging for tracking changes and understanding the impact of modifications. An interactive improvement process based on performance evaluation helped refine the model architecture and training regiment. Versioning and logging are essential tools for the development of a software model.

4.4.2 Model Limitations

The current model, while effective for the puzzle-solving task, demonstrates several limitations when considering its application to the domain of molecular recognition.

The depth of the model used in the jigsaw puzzle solver is relatively shallow. This simplicity, although sufficient for the structured and somewhat predictable domain of jigsaw puzzles, falls short of capturing the intricate and highly variable nature of protein interactions. A deeper and more complex model, such as a Transformer-based architecture, would be more suitable for protein docking due to its ability to capture long-range dependencies and intricate patterns in large-scale data.

Transformers, renowned for their effectiveness in handling sequential data in natural language processing, could potentially enhance the performance in protein docking. Their architecture is capable of capturing long-range interactions, enabling the model to learn amino acid relationships and more complex interactions over greater distances than typical CNNs. This capability is essential in molecular recognition, where the complex interplay between distant amino acids can influence the binding site.

A major hurdle in applying advanced machine learning techniques like those used in the AI-Solver to protein docking is the availability of extensive, well-annotated training datasets. Protein interactions are notoriously difficult to label accurately, requiring extensive experimental work that is both time-consuming and costly. This limitation restricts the amount of data available for training more complex models, which in turn can impede the model's ability to learn.

With the increase in the size of the model required by the complexity of the problem, so does the demand for computation resources. Larger models require significant amounts of memory and processing power, often necessitating the use of distributed systems. This increase in computational demands can lead to higher costs and longer training times, which may be a limiting factor in research settings.

4.5 Future Directions In Molecular Recognition Modeling

The rapid evolution of computational methods in protein design, especially highlighted by the advent of AlphaFold, has ushered in a new era of possibilities. AlphaFold's impact on drug design is profound, providing a paradigm shift that could dramatically accelerate phases of the drug discovery pipeline.

4.5.1 AlphaFold and Molecular Recogntion

AlphaFold has revolutionised the approaches and understanding of protein structures. The detailed and accurate protein structures predicted by AlphaFold provide a robust basis for exploring protein-protein interactions. Utilising AlphaFold's capabilities, researchers can potentially transform the initial phases of the drug discovery workflow into an automated, highly accurate process.

The integration of AlphaFold's structure predictions can be leveraged to produce precise embeddings of protein complexes. These embeddings can then serve as the foundation for a computational model designed to score the interactivity between potential drug molecules and their target proteins. Inspired by the pertaining approach used in the jigsaw puzzle solver, where segmentation aids in accurate piece identification, AlphaFold's embeddings could similarly enhance the accuracy and speed of hit identification and lead optimisation phases. By applying a fine-tuned head to these embeddings, the model could directly predict interaction hotspots, significantly reducing the time and computational resources required for finding and screening potential hits.

4.5.2 The Next Frontier: Molecular Recognition Design

The precise structural insights provided by AlphaFold enable a new approach to drug design, Molecular Recognition Design (MRD). MRD aims to streamline the early stages of drug discovery, particularly targeting the identification and optimisation phases. With accurate structural prediction at hand, MRD utilises generative modelling techniques to propose novel protein designs or small molecules that are tailored to interact with specific targets.

In MRD, once a target protein is identified, generative models can be employed to design new ligands or modify existing proteins to enhance their binding affinity and specificity. These models, trained on vast datasets of protein interactions and their outcomes, can generate new, or modify existing proteins to improve drug efficacy and reduce side effects.

By integrating MRD into the drug design workflow, the process from hit identification to lead optimisation could be significantly condensed. This approach not only speeds up the discovery cycle but also enhances the precision of targeting mechanisms, potentially leading to more effective therapeutics with quicker development timelines.

The future of molecular recognition modelling holds promising potential for transforming drug design. By exploiting breakthroughs like AlphaFold within a framework inspired by computational puzzle-solving, researchers can accelerate drug discovery and develop improved therapeutic interventions.

4.6 Conclusion

This chapter delves into the analogy between assembling jigsaw puzzles and the complex domain of molecular recognition in protein design. By exploring this simplified process, the strategies employed for jigsaw puzzles can illuminate potential methods in molecular docking and drug discovery. Both processes depend on accurately identifying and aligning interacting parts, puzzle pieces or molecular structures.

Utilising deep learning, particularly convolutional neural networks, the feasibility of segmenting and interpreting intricate datasets analogous to discerning active sits in proteins was demonstrated. This capability underpins more precise drug design methodologies, where predicting molecular interactions can expedite the development of targeted therapeutic interventions. Using innovative synthetic data generators to train these models showcases the importance of robust datasets, which help overcome the limitations posed by scarce experimental data in drug discovery.

The application of AI solvers, initially designed for jigsaw puzzles, to molecular recognition tasks, illustrates a promising approach to accelerating drug design processes. The techniques developed have the potential to significantly reduce the time and resources needed from hit identification to lead optimisation.

The insights gained from applying machine learning techniques to jigsaw puzzle solving have broader implications for computational biology and drug discovery. This interdisciplinary approach not only offers a fresh perspective but also highlights the utility of AI in scientific research and application, paving the way for future innovations in protein interaction studies.

Chapter 5

Parametric Protein Design, Silver Surfer & PS-GO

This chapter examines the cutting-edge strategies in parametric protein design, specifically through the developments of Silver Surfer [194], a platform for parametric protein design, and PS-GO [200], a parametric protein search engine. It reviews the particularities of both solutions and concludes with a comparative analysis of the two models.

5.1 Introduction

The relentless progress in computational methods and biotechnology has propelled the frontiers of protein design into a new era, where data-driven, *in silico* approaches are increasingly accurate. Following the detailed discussion in previous chapters on protein folding and docking, this chapter transitions towards the innovative domain of parametric protein design. The chapter delves into the advanced methodologies and technologies spearheaded by Silver Surfer and PS-GO, which are at the forefront of parametric protein design. This chapter aims to harness the explosive growth of protein-related data, integrating diverse data sources into a unified search portal that mirrors the functionality of search engines.

The recent explosion of protein property prediction models presents both a challenge and an opportunity. This chapter investigates methods of leveraging predictive models that determine specific protein properties based on their amino acid sequences [85, 95, 151, 162, 170] with a focus on parametric searches that locate end even dynamically generate proteins with desired characteristics.

Central to the exploration is the application of genetic algorithms for parametric protein generation, a shift from mere data retrieval to sophisticated, generative techniques, capable of navigating the immense protein space. This approach aims to demonstrate the feasibility of employing genetic algorithms to explore the immense protein space. This chapter sets the stage for generative drug design, a promising field where the principles of parametric protein design are applied to develop new therapeutics. Four predictors of general utility in the field of bioinformatics are used

for demonstrative purposes throughout the chapter. The four properties are: Instability Index, Mono-Isotopic Mass, Grand Average of Hydrophobicity (GRAVY), and Isoelectric Point;

5.1.1 Theoretical Background

The recent surge in protein-related data holds illuminating powers for the fields of bioinformatics and computational biology. This data encompasses a wide array of protein structures, functions, and genetic variations, necessitating a method for effective retrieval and comparison in order to facilitate a comprehensive understanding. These needs echo the challenges once faced by the early Internet, where the rapid expansion of online content demanded robust solutions for data retrieval and organisation.

A paradigm-shifting solution in the realm of Internet technology was the introduction of Google's PageRank algorithm [36]. This algorithm revolutionised information retrieval by using the hyperlink structure of the web to rank pages based on their importance and relevance, fundamentally changing how data was accessed and utilised online. Drawing inspiration from this, there is a pressing need to adopt a similar transformative approach to manage the gigantic volumes of protein data in the biosciences. The ideal solution should reveal the structure of proteins, a key determinant in their function and interaction, and enable precise searches and comparisons within expansive datasets. This solution must integrate the latest advances in computational biology and data science to handle, analyse, and interpret big data efficiently.

The PS-GO search engine was developed in response to this challenge as a pioneering approach to protein data exploration. PS-GO leverages computational models and algorithms to enable robust data handling, accurate predictions, and detailed comparisons of protein structures and functions. This engine exemplifies how technological innovation can bridge the gap between data availability and usability, much like search engines have done for the nascent Internet.

At the core of understanding protein function and design is the concept of protein conformation, the spatial and structural arrangement of amino acids within a protein. The protein's conformation is crucial because it dictates the proteins' stability and their functional interactions. By studying the link between structure and function, scientists can design better farmaceuticals. This deep understanding facilitates the application of parametric protein search and design, where specific features and functions are engineered based on desired outcomes.

5.1.2 Parametric Protein Design

Parametric Protein Design (PPD) represents a transformative approach within the field of Computer Aided Drug Design (CADD), where the objective is to engineer

or modify proteins to obtain desired functions, stability, and solubility through computational manipulation of amino acid sequences. This represents a paradigm shift from the traditional approach of testing massive amounts of proteins for activity. This shift can be conceptually expressed in common experience as the shift from searching for pieces of pre-made furniture and hoping for a good fit within the existing layout of a room to ordering custom furniture tailored to fit specific spatial and functional needs. The goal is to specify requirements upfront and initiate a targeted search for proteins that meet these specifications.

PPD is a proven approach within CADD, significantly impacting the pharmaceutical industry. In 2018, eight of the top ten best-selling drugs were based on proteins that have been manufactured in, extracted from, or semi-synthesised from biological sources [216]. A substantial part of their success can be attributed to their high specificity, affinity, and reduced toxicity compared to small molecule drugs [43, 147]. Traditionally, the process involved discovering functional sequences in nature and modifying them to express desired properties. However, this method presents several drawbacks, such as the laborious and expensive identification process and the dependence on naturally evolved proteins, which may not align with therapeutic needs due to evolutionary constraints. Furthermore, the evolutionary process is slow, having explored only a fraction of the possible proteins [74, 306] and may be lagging behind in our fast-evolving and rapidly changing environment [74].

One of the challenges of PDD is represented by its vast search space, given the exponential number of combinations of amino acids. Furthermore, many of these combinations result in proteins that either do not fold into stable configurations or lack biological functionality. This challenge can be considered analogous to language generation, where random combinations of letters rarely form coherent words; similarly, random sequences of amino acids are unlikely to yield functional proteins. The major distinction between language generation and protein sequence generation is that the rules for protein generation are more complex, and the "words" usually span hundreds of characters.

Two of the main steps in PDD are parameter selection and parameter optimisation. Parameter selection entails choosing attributes relevant to the design goals, such as folding stability or ligand affinity. Parameter optimisation may involve strategies like energy minimisation, focusing on optimising the energy of a cost function, or probabilistic search techniques such as Monte Carlo sampling, which explores the parameter space to find viable protein configurations.

Understanding the intricate relationship between a protein's structure, parameters, and function is fundamental in PPD. By altering the amino acid sequence, one can modify properties such as fold stability and functional properties; adjusting functional properties like solvent accessibility, the affinity of a protein to a specific ligand can be targeted, enabling more precise drug design.

PDD has already facilitated significant advancements in various applications, including enhancing enzyme activity, improving antibody affinity, and designing

novel functional proteins [8]. Its potential to revolutionise fields like biotechnology, environmental sciences, and pharmaceuticals is immense. In the pharmaceutical sector, for instance, PDD can lead to the development of efficient, sustainable [68] targeted therapeutic agents [309], accelerating drug discovery and reducing associated costs.

5.1.3 Challenges posed by searching the protein space

The task of exploring and understanding the protein space encompasses numerous challenges, primarily due to the reliance on extensive experimental data stored across various databases. Key resources such as the RCSB Protein Data Bank [17], a widely-used repository for 3D protein structures determined by experimental methods, along with UniProt [60], which integrates protein sequences and functional annotations from several databases such as CAT [222], SCOP [215], and Pfam [93] categorise proteins based on their structural, sequence, or functional characteristics, which are indispensable for research and design in protein-related fields.

Protein searching primarily employs two principal approaches: structure-based and sequence-based. Structure-based methods, including DALI [126] and TM-align [337], compare the 3D structures of proteins to identify similarities, particularly useful for proteins that exhibit structural similarities despite diverse sequences. Conversely, sequence-based methods like BLAST [327], and HMMER [92] excel at detecting homology by comparing amino acid sequences.

These methods face limitations in terms of efficiency, accuracy, and adaptability [105, 254, 270]. Structure-based methods are computationally intensive and slow, which can be a significant hurdle when dealing with large datasets. Sequence-based methods may struggle with proteins with considerable structural differences despite sequence similarity, as the complex relationship between protein structure and sequence can lead to errors in search results. Additionally, inaccuracies such as incorrect annotations, data redundancy, and incomplete data can further negatively impact search results.

To address these issues, a series of solutions have been developed for advanced protein structure search incorporating modern computational techniques. 3D-AF-SURFER [1] utilises deep learning to learn 3D feature representations of protein surfaces, enabling fast and accurate structure alignment and similarity search. GraSR [319] employs graph representation learning, improving search efficiency and generalisability by converting protein structures into graphs and learning their embedded representations. DeepFold [171] uses convolutional neural networks to extract features directly from 3D structures, accelerating the structure alignment process. Despite these advancements, challenges remain in terms of insufficient utilisation of sequence information, limited search flexibility, and interoperability.

Parametric protein search presents a promising alternative that addresses these challenges from a different perspective. By leveraging the principles of parametric

protein design, this approach selects and optimises a set of key parameters that define protein structure and function, enabling a more comprehensive and fine-grained representation of protein. This method seamlessly integrates sequence and structural information, capturing both the global conformation and the local physicochemical properties, and offers a highly flexible and customisable framework.

5.1.4 Beyond the Known Protein Space

The landscape of proteins, as explored through historical drug discovery processes, presents a distinctly fragmented, sparsely populated, predominantly clustered around naturally evolved proteins aspect. This clustering limits the scope of exploration to a narrow window of protein functionalities that nature has optimised for survival rather than specific human therapeutic needs. Addressing this limitation necessitates a strategic departure from traditional methodologies to embrace innovative approaches capable of navigating and extrapolating the known protein space.

The use of genetic algorithms for parametric protein generation, as demonstrated by tools like SilverSurfer, exemplifies such a promising approach. Genetic algorithms have been chosen for this task due to their robust capability to mimic the complex evolutionary process in an artificial context. They excel in exploring the vast and complex protein sequence space by iteratively generating and selecting sequences based on their fitness, aligned with predetermined functional targets. This iterative nature allows for the discovery of novel amino-acid configurations that may not have been naturally selected but could be highly beneficial for specific applications [146].

The generative potential of genetic algorithms to create new proteins is transformative, offering prospects for significant advancements across various domains. These include the development of innovative treatments for diseases, where novel proteins could serve as therapeutics or as targets for drug development. Similarly, the synthesis of sustainable bio-fuels and the engineering of bio-materials could benefit immensely from proteins designed to exhibit specific catalytic or structural properties that enhance efficiency and functionality [68].

Transcending the boundaries of the known protein space, the parametric protein design facilitated by genetic algorithms has the potential to broaden the horizons of bioinformatics and molecular biology to catalyse the transition from depending on natural evolution to designing targeted proteins. This shift holds the promise of accelerating the discovery and implementation of protein-based solutions to some of the most pressing challenges in medicine, energy, and materials science.

5.2 PS-GO: Protein Search

The evolution of computational biology has ushered in sophisticated tools capable of transforming the way scientists engage with protein data. Among these innovations, PS-GO stands out as a SOTA parametric protein search engine. The platform, depicted in Fig. 5.1, integrates an extensive array of functionalities for protein search, analysis, and visualisation through a user-friendly web interface. PS-GO enables intricate searches across protein databases by allowing the users to define search criteria based on a combination of sequence, structure, and physicochemical properties. This flexibility is further enhanced by a parameter selection tool and the capability to input search queries through natural language descriptions.

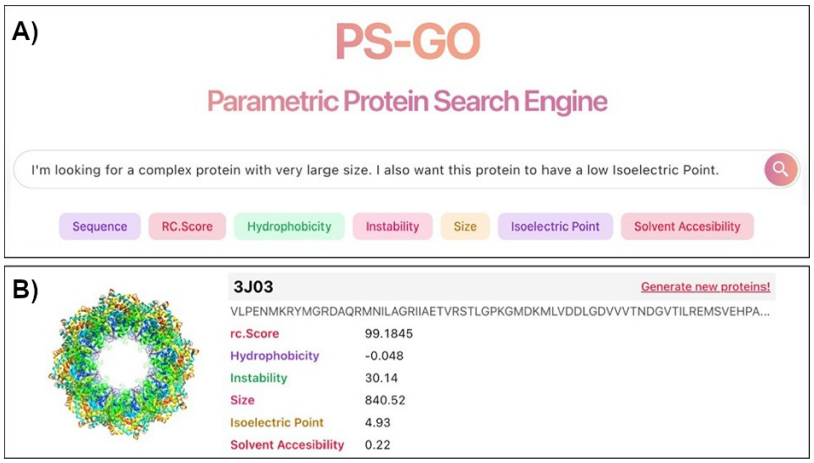


FIGURE 5.1: PS-GO protein search page; A. - Search component, B. - Results list component. Source: Mi et al. [200].

The interface of PS-GO is laid out to maximise usability and accessibility. It features a comprehensive suite of functionalities that allow for the detailed search, analysis and visualisation of proteins. This is accessible through a streamlined web interface where users can easily specify search criteria. The search panel, shown in Fig. 5.1 A, offers a variety of filters. The results of these searches are displayed in a clear, concise manner, as shown in Fig. 5.1 B. Each result provides essential information at a glance, including the protein's model visualisation, sequence, and a comprehensive set of physicochemical properties. This allows users to quickly assess the relevance of each result.

Designed to overcome the limitations of traditional protein search tools, PS-GO incorporates protein structure and sequence information with computable parameters to provide a comprehensive search and analysis framework. This integration addresses critical gaps in existing methods by offering enhanced interpretability through detailed parameter profiles and advanced visualisation features.

A foundational functionality of PS-GO is its unique parametric search capability, which allows users to precisely control and define the functional parameters of their searches. This feature is instrumental in refining the search process, ensuring that the

results are not only accurate but also relevant to the specific scientific enquiries. PS-GO supports a range of parameters, including RC. Score, Hydrophobicity, Instability, Size, Isoelectric Point and Solvent Accessibility, each of which can be fine-tuned to meet the user's specific functional requirements.

Enabled by the power of parallel computing, PS-GO is capable of handling extensive datasets and complex queries with efficiency. The platform's customisable nature allows users to modify parameters and optimisation conditions to fit their specific experimental and research needs. In conjunction with the SilverSurfer service, PS-GO empowers researchers to delve deeper into the protein sequence space, exploring and designing proteins with unprecedented precision.

PS-GO also integrates with the Protein Fragment And Structure Analysis (PROFASA) platform, enhancing its utility by providing additional layers of structural insights and predictive capabilities. The combination of these tools encapsulates the cutting-edge of computational biology, offering a holistic approach to protein research and paving the way for innovative discoveries in the field.

5.2.1 Parametric search strategy

PS-GO employs a novel parametric search strategy designed to efficiently navigate and explore the vast protein sequence space. This strategy utilises a system that optimises the search and retrieval of protein data based on multiple functional parameters. Here, we detail the approach adopted by PS-GO for handling, processing, and searching through protein-related information.

Data Ingestion

The ingestion phase of the PS-GO model involves the systematic collection and processing of protein data from authoritative sources such as the PDB and UniProt. This step ensures the data utilised is of high quality and relevance. Automated scripts fetch protein structure and sequence files in mmCIF and FASTA formats, respectively, along with XML format annotation flies from UniProt for curation. Data undergoes rigorous curation to filter and validate sequence and structural integrity, remove redundancy, and ensure completeness. Tools from the BioPython library parse the necessary fields, which are then mapped into a relational database schema designed for quick access and complex queries. An advanced indexing and hashing mechanism is implemented, where each protein's calculated parameters are categorised into hash bins, allowing for rapid retrieval from the database. This involves multi-dimensional hashing for numerical data and MurMurHash3 [66] for categorical data, ensuring minimal collision.

Parameter Calculation

Comprehensive parameters are computed for each protein entry to characterise its physicochemical properties and structural features. These include amino acid composition, hydrophobicity, instability index, molecular weight, isoelectric points, secondary structure information and solvent accessibility. These parameters are crucial for the subsequent steps of integration and search, providing a detailed profile of each protein that aids in precise query matching.

Data Integration

The integration of stored parameters with existing sequence and structural data forms the backbone of PS-GO's parametric search capabilities. During integration, data is meticulously indexed and optimised within a MySQL relational database, leveraging the database/SQL packages in Go. This allows for efficient data handling and retrieval, necessary for managing complex queries on large datasets.

Parametric Search

PS-GO performs dynamic parametric searches based on user-defined criteria. Users can input queries specifying desired protein characteristics such as sequence similarity, structural attributes, or specific physicochemical properties. PS-GO's indexing system quickly identifies and retrieves proteins that match these specifications. The system also offers the possibility to express the query using natural language and interprets the queries using OpenAI's GPT-4, translating them into structured search queries. For compound queries, it processes each condition separately, aggregates the results, and applies intersection operations to ensure that all specified conditions are met. Search results are then ranked according to their relevance, providing users with proteins according to their research needs.

Beyond searching, PS-GO supports the design of novel proteins through its integration with the SilverSurfer backend. Users can specify target parameters, and PS-GO, through its integration with the SilverSurfer, can generate new protein sequences that meet the desired criteria.

5.3 SilverSurfer

The SilverSurfer service stands as a pioneering platform in the realm of parametric protein design, distinctly crafted to navigate the vast and intricate protein space. SilverSurfer is developed to facilitate the discovery and engineering of novel proteins, operating on the foundation of a custom-designed genetic algorithm tailored to handle the variability inherent in protein sequences.

Central to the functionality of SilverSurfer is its ability to search the protein space starting from one or more seed sequences. These sequences act as the genetic material for a series of evolutionary iterations aimed at discovering proteins that meet the

predefined functional criteria. With the help of the mutation operator, SilverSurfer generates a diverse pool of sequence candidates that explore the protein space. Each candidate is evaluated through a fitness function which assesses their suitability against the desired properties.

The SilverSurfer generation process is dynamic and iterative, continuing until SilverSurfer successfully identifies the required number of proteins that align with the user's specifications or exhausts the protein space. The generation is designed to be exhaustive yet efficient, ensuring a thorough exploration of the protein space while optimising for paths with higher promise. This makes SilverSurfer an invaluable tool for researchers aiming to push the frontiers of bioinformatics and pharmacological innovation through parametric protein design.

5.3.1 Genetic Algorithms Particularities

In the context of the SilverSurfer service, genetic algorithms are harnessed to navigate the protein sequence landscape efficiently. This utilisation aims to capitalise on their ability to generate diverse protein variations through evolutionary-inspired operations. The key to the success of any genetic algorithm is the strategic choice of encoding, genetic operators, and fitness functions.

Encoding

Choosing the right encoding strategy is crucial for the effective implementation of any Genetic Algorithm. In the case of the Silver Surfer, protein sequences are encoded using the conventional one-letter representation of amino acids. This encoding strategy is deeply rooted in standard bioinformatics practices, ensuring that the platform remains relevant and compatible with future scientific studies. This choice aligns with current methodologies and ensures the long-term viability of the platform.

The adoption of this robust encoding system allows for the seamless integration of various protein sequence property predictors. These predictors can readily interpret the encoded data, enhancing the platform's capability to assess and optimise the protein sequences dynamically. By leveraging this encoding format, SilverSurfer maintains a balance between computational efficiency and complexity necessary to explore the vast protein design space effectively.

Mutation and Crossover

Mutation and crossover are essential operations in evolutionary algorithms that introduce and recombine genetic variations, driving evolutionary progress. The SilverSurfer platform utilises these genetic mechanisms for the exploration of the protein space, aiming to generate novel proteins that meet predefined functional goals.

Mutation serves as a primary method for introducing genetic diversity by altering genetic sequences. SilverSurfer employs multiple mutation strategies to balance

the exploration and exploitation phases within its evolutionary approach. The first strategy, known as a random mutation, involves the alteration of an amino acid in a sequence to another amino acid chosen at random. For example, changing 'C' to 'G' in the sequence 'DSACGA' results in the sequence 'DSAGGA'. This method introduces unpredictability and diversity into the population.

A more refined approach in the SilverSurfer is the contribution-based selection strategy for mutation. This method calculates the individual contribution of each amino acid to a targeted property, such as monoisotopic mass, and selectively mutates the amino acid that has the most significant impact on achieving the desired property value. For instance, if the sequence 'DSACGA' has a total monoisotopic mass exceeding the target of 460, the algorithm identifies the amino acid with the highest individual mass contribution and selects it for mutation. This approach allows for a more focused search by directing mutations towards those changes that most effectively optimise the desired property.

The crossover operation combines the genetic information from two parent organisms to produce offspring that inherit beneficial genes from both of the parents. In SilverSurfer, the crossover is implemented to respect user-defined 'frozen sections', specific segments of the protein sequences that are marked by the user as crucial and essential not to be altered during the genetic operations. By preserving these sections, SilverSurfer ensures that essential characteristics of the proteins are maintained.

Fitness

The efficacy of SliverSurfer's genetic algorithm is heavily dependent on the performance of its fitness function, which must evaluate how closely a protein sequence aligns with predefined objectives. In dealing with protein design, SilverSurfer confronts the challenge of complex interdependencies among various protein properties. These properties typically lack straightforward trade-offs, making traditional Pareto optimisation approaches unstable. SilverSurfer employs a normalised absolute distance metric as its fitness function. This metric computes the sum of the absolute differences between the observed and targeted values of each protein property, which is then normalised to account for variations in property scales. This method not only simplifies the assessment of protein fitness by consolidating multiple property evaluations into a single scalar value but also maintains the simplicity of computation required for the massively multiple evaluations required by any evolutionary approach.

5.3.2 Implementation

The implementation of SilverSurfer uses Python 3.9, a programming language renowned for its versatility and broad acceptance within the bioinformatics community. This choice is anchored in Python's well-established ecosystem, which supports rapid

prototyping and offers extensive libraries specialised in bioinformatics. These libraries can facilitate the manipulation of protein sequences and their associated properties, streamlining the development of additional protein predictors.

SilverSurfer's design emphasises customisation, allowing users to tailor the genetic algorithm's operation to fit their specific research needs. Users can modify various aspects of the algorithm, including the selection of mutation sites and the application of different mutation strategies. This adaptability extends to the assignment of 'freeze masks', constraints that prevent certain segments of the protein sequence from being altered during the application of the genetic operators. Additionally, users can set a dissimilarity threshold to ensure a diversity of results, signalling to the algorithm the need for a more varied result set.

The architecture of SilverSurfer is engineered with extensibility as a core tenet, easily accommodating the addition of new properties or behaviours. This flexibility makes it an ideal platform for probing the vast protein space, offering researchers a powerful tool for designing proteins with specific physicochemical properties. By integrating these capabilities, SilverSurfer stands out as a robust instrument in the field of parametric protein design, pushing the boundaries of what is achievable in computational biology and pharmacological research.

5.3.3 Protein Generation

The protein generation process within the SilverSurfer platform is underpinned by a customised genetic algorithm that efficiently explores the vast protein space. This subsection details the robust, iterative process designed to generate novel proteins that meet specific predefined criteria.

Initialisation and Iterative Search

The SilverSurfer's genetic algorithm is initialised when a user supplies a protein sequence. Suppose the user does not desire to submit a starting seed to focus the search. In that case, SilverSurfer can generate a random sequence or utilise PS-GO to search for the closest proteins to the given targets, using them as seed sequences. Alongside the initial sequence, target value and their acceptable ranges are defined within the acceptance component.

Once initialised, the searcher works iteratively, processing sequences from an unexplored priority queue. This process continues until the required number of protein sequences that pass the acceptance threshold is identified or until the queue is depleted. During each interaction, the best-scoring protein is extracted for further processing.

Genetic Operators and Evaluation

Genetic operators are applied according to the selected strategy, respecting any defined 'frozen' segments that should not be altered. These operators generate a new

set of protein sequences, which are then filtered through a bloom filter to exclude any previously processed candidates. Subsequently, each candidate's relevant physicochemical properties are computed and evaluated by the acceptance component.

The acceptance component plays a crucial role during this phase, defining the goal; its purpose is to evaluate each sequence to determine if it falls within the preset acceptance criteria. Sequences that meet these criteria are added to the results list.

To manage memory efficiently and maintain a high level of performance, SilverSurfer implements a population threshold limit based on the available memory. When this threshold is reached, a clustering algorithm is applied to the lower-performing individuals in the queue. These individuals are replaced with centroid sequences, reducing the population size while minimising the informational loss.

SilverSurfer's implementation is highly parallelisable, orchestrated by the priority queue, which can synchronise the computation tasks across multiple threads. This design allows for the simultaneous computation of physicochemical properties for each sequence, optimising the use of multithreaded environments.

The fitness of each protein sequence is evaluated based on the normalised absolute distance from defined targets, incorporating four properties of general utility in the bioinformatics field as a proof of concept.

$$\sum_{i=1}^{n} Mass(A_i) - (n-1) \times 18.01528\text{Da} \tag{5.1}$$

Monoisotopic Mass is computed according to the 5.1 equation, summing the standard atomic masses A_i of each amino acid in the sequence and adjusting for the mass of water molecules, $n-1 \times 18.01528$Da lost during peptide bonding.

$$pH = pK_a + log_{10}\left(\frac{[Base]}{[Acid]}\right) \tag{5.2}$$

Isoelectric Point is determined using the Henderson-Hasselbalch 5.2 equation, with pK_a values from Wikipedia [240], values selected due to their RMSD scoring among the publicly available pK_a values, according to IPC [154]

$$\frac{10}{n}\sum_{i=1}^{n-1} InstabilityScore(A_i, A_{i+1}) \tag{5.3}$$

Instability Index is obtained according to the 5.3 equation; evaluated using a direct implementation of the Guruprassad et al. [111] paper. The index is computed by summing the bi-peptide instability scores $InstabilityScore(A_i, A_{i+1})$ across the protein and scaling the value according to its length.

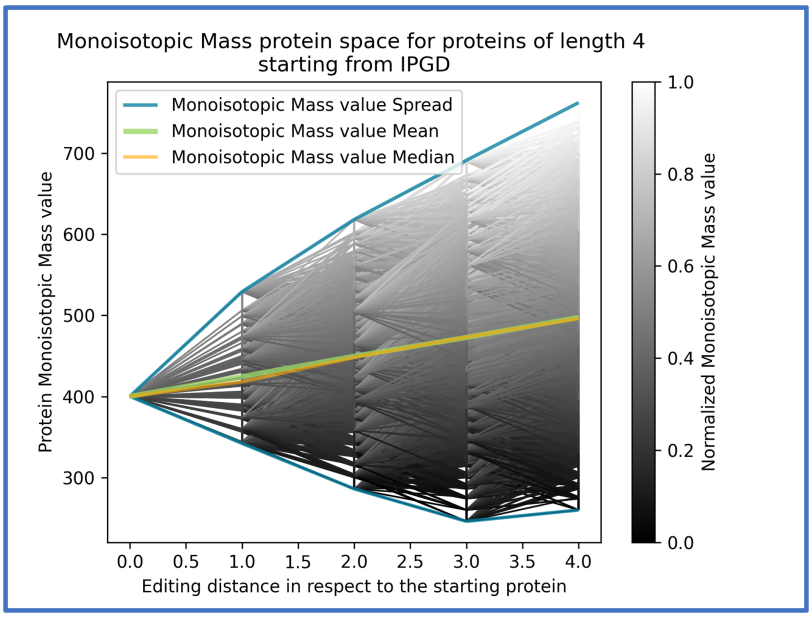


FIGURE 5.2: Exploration of the Monoisotopic Mass protein space.

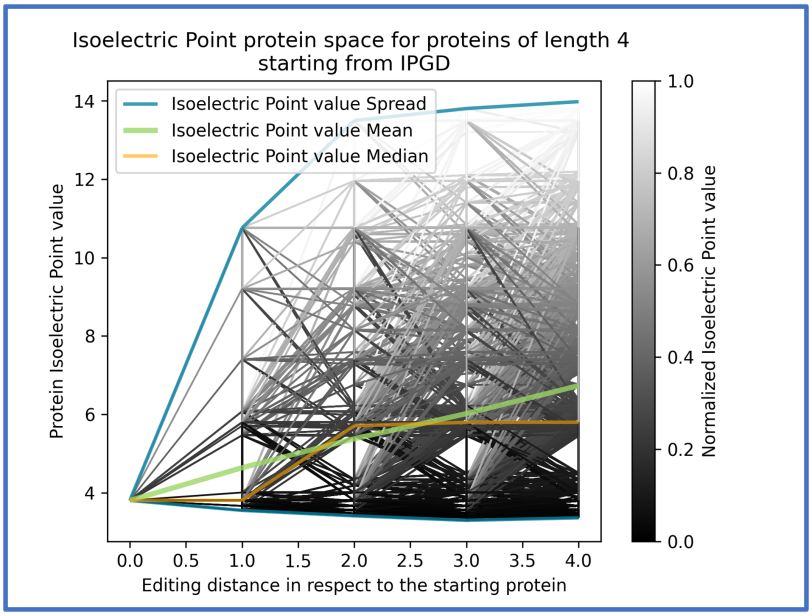


FIGURE 5.3: Exploration of the Isoelectric Point protein space.

$$\frac{1}{n}\sum_{i=1}^{n} Hydropathy(A_i) \tag{5.4}$$

GRAVY score , a measure of protein hydrophobicity, is calculated as illustrated by the 5.4 Equation. The GRAVY score is obtained by averaging the hydropathy, $Hydropahty(A_i)$, values of all amino acids in the sequence.

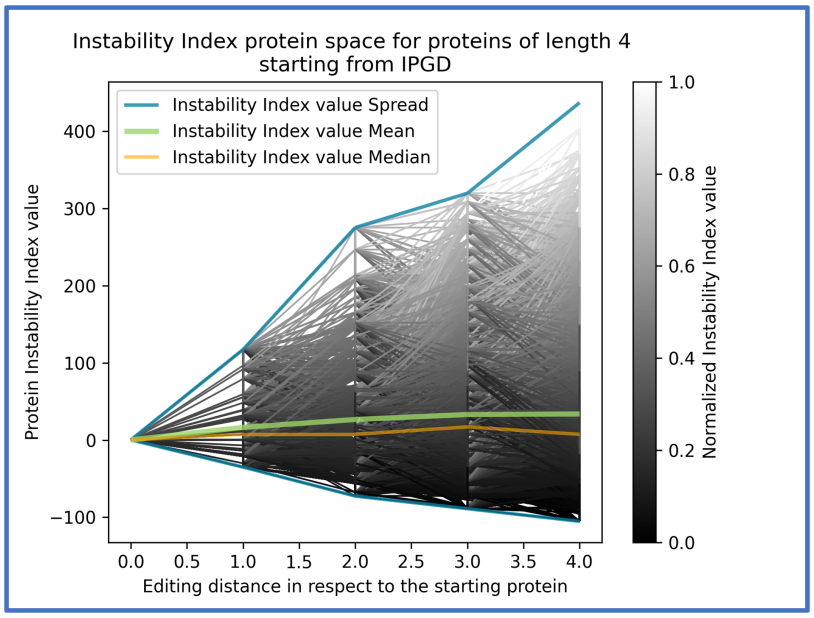


FIGURE 5.4: Exploration of the Instability Index protein space.

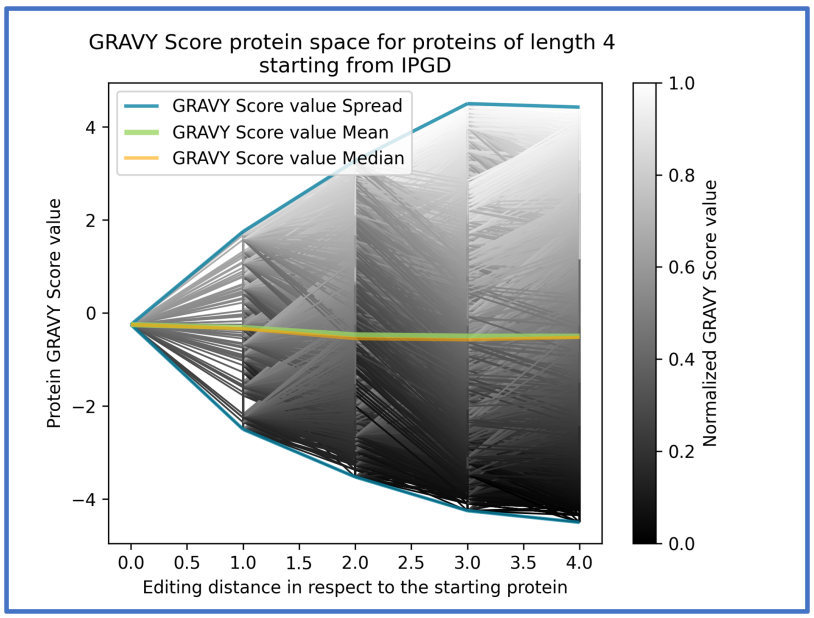


FIGURE 5.5: Exploration of the GRAVY protein space.

All the properties currently in use by the SilverSurfer backend are calculated using hand-coded implementations of well-established algorithms, ensuring that SilverSurfer can accurately evaluate the fitness of protein sequences against the desired characteristics.

5.3.4 Results

Validating the properties

The SilverSurfer platform was rigorously validated to ensure the accuracy and reliability of its protein property predictions. The validation process involved a systematic comparison of the hand-coded algorithms against established web tools that specialise in calculating these protein properties. This comparison provided a critical assessment of the reliability and accuracy of the protein property predictors.

Monoisotopic Mass validation involved comparisons with data from the Swiss Institute of Bioinformatics' PeptideMass [101], Prot PI [238] and the Protein Calculator [239]. These comparisons confirmed that Silver Surfer accurately computes the mass of proteins based on their amino acid composition.

Isoelectric Point ; critical for understanding protein solubility and charge state under various pH conditions, was validated by comparing SilverSurfer's results with those from ProtParam [101], The Isoelectric Calculator [154] and Prot Pi [238]. These comparisons confirmed the accuracy of the isoelectric point prediction.

Instability Index ; a measure of protein stability was validated against results from the Guruprasad et al. paper, which introduced this index [111], the Swiss Institute of Bioinformatics' ProtParam [101] and the Peptides R package [224]. The close alignment of these results demonstrates the accuracy of the SilverSurfer.

GRAVY Score determines the overall hydrophobicity or hydrophilicity of a protein, impacting its solubility and interaction with other molecules. SilverSurfer's GRAVY scores have been validated against outputs from the Swiss Institute of Bioinformatics' ProtParam [101], the Alakazam R package [109], and the Gravy Calculator [98]. The results confirm that SilverSurfer's GRAVY score is accurate.

The results of the validation process confirm the robustness of SilverSurfer's computational approach and validate its accuracy.

Parametric Generation

The Parametric Generation method used by SilverSurfer leverages the generative powers of genetic algorithms in order to propose sequences with properties that fall within the user-specified criteria. For the evaluation of the generative capabilities of the Genetic algorithm, a normalised absolute distance was used as a scoring function; an acceptance function with an epsilon deviation of maximum $1e10^{-14}$ from the stated property value; a contribution-based mutation strategy with a dynamic mutation rate; an empty freeze mask and a requiring one result.

The validation was conducted by selecting two different sequences, one to serve as the starting sequence, and one for which the properties are computed and used

as target values. The computational analysis covered the entire protein space for sequences of length 5, a length considered to be of sufficient size to be representative of the complexity while still manageable within the hardware constraints. The exhaustive evaluation involved the computation of properties for 3,200,000 unique protein sequences.

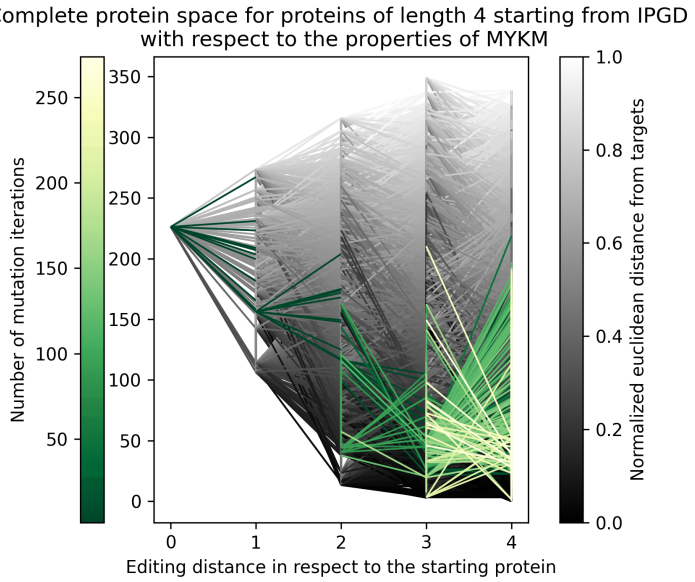


FIGURE 5.6: Illustrative example of the search path undertaken by the SilverSurfer in the complete protein space of proteins with a length of 4 amino-acids.

Figure 5.6 visually summarises the mutation search path undertaken by SilverSurfer; it illustrates the mutation distance from the starting sequence on the x-axis against the distance from the target properties on the y-axis; the figure employs a grayscale colouring of the connections to display the normalised Euclidean distance from the target values. The color-highlighted path shows trajectories that SilverSurfer explored, with those closer to the x-axis being closer to the correct sequence. The figure illustrates the order in which the sequences have been explored with a gradient from green to light yellow. This graphical representation highlights SilverSurfer's efficient search focus, directing computation efforts towards promising regions of the protein space.

The validation process demonstrated SilverSurfer's capability to navigate the protein sequence space in order to generate a protein with the desired requirements, ensuring that, if a feasible solution within the defined constraints exists, it will be discovered. This robust approach showcases the potential of SilverSurfer in the Parametric Protein Generation paradigm.

5.4 Comparative Analysis

The advent of sophisticated computational models has ushered in a new era in protein design, characterised by the simultaneous development of multiple methodologies aimed at enhancing the precision and efficiency of protein searches. This section conducts a comparative analysis of the distinct methodologies employed by PS-GO and SilverSurfer.

5.4.1 Unique Approaches of PS-GO

PS-GO exemplifies a strategic focus on parametric design by implementing a system that is tailored to meet the dynamic needs of bioinformatics. Its design philosophy combines computational biology with cutting-edge data science techniques to enhance the efficiency and effectiveness of protein searches.

PS-GO improves protein structure design by focusing on a parametric approach that goes beyond traditional sequence alignment and structural comparison. It leverages computational models to facilitate the search for proteins with specific functional properties and stability. This approach allows researchers to specify exact parameters as the foundation for searching protein sequences. Such specificity is particularly advantageous in designing proteins for therapeutic purposes, where the stability and function of the protein directly impact its efficacy and safety.

At the core of PS-GO's effectiveness lies its robust data management system, which enables PS-GO to integrate diverse datasets spanning protein sequences, structures, and functional annotations. PS-GO uses data ingestion and integration techniques to ensure that all available proteins are complete and usable for parametric search. This system is built to handle large volumes of data efficiently, reducing the time and computational resources needed to find and compare protein structures. By employing advanced indexing and hashing strategies, PS-GO enhances the speed of data retrieval while painting the level of accuracy, making it an invaluable tool for researchers navigating the expansive protein landscape.

5.4.2 Unique Approaches of SilverSurfer

SilverSurfer stands out in the field of parametric protein design for its innovative use of a customised genetic algorithm tailored to navigate the complex landscape of protein sequences. This approach harnesses the power of evolutionary principles to explore and generate protein variants that might not be accessible through historical methods.

One of the foundational pillars of SilverSurfer's strategy is its usage of a genetic algorithm that adapts to the complexities of protein design, a customised solution capable of handling the diversity and complexity inherent in protein structures. By stimulating the natural evolutionary process, SilverSurfer explores a broader range of potential protein configurations, making it an invaluable tool for researchers.

The SilverSurfer system employs dynamic genetic operators that are crucial for refining the exploration of protein sequence space. These operators are strategically applied based on the contribution of each amino acid to the target properties. This targeted approach allows SilverSurfer to focus on mutations that are more likely to yield beneficial properties, thereby enhancing the efficiency of the genetic algorithm in finding viable protein solutions.

The selection process prioritises sequences that demonstrate the greatest potential to meet specific functional requirements. This is achieved through a combination of the fitness function coupled with the priority queue that ensures that individuals with the most promising genes get to spread their genetic material more often.

Another of the foundational pillars of SilverSurfer's philosophy is the high degree of customisability. Researchers can tailor the genetic algorithm to suit specific scientific or experimental needs by adjusting various parameters, such as mutation rates, selection criteria, or the handling of genetic operators. This flexibility extends to the incorporation of 'freeze masks', which allow certain segments of the protein sequence to remain unchanged during genetic operations, preserving essential functional regions while exploring new variations. SilverSurfer is designed to accommodate the integration of new predictor models, ensuring that it remains at the cutting edge of technology in protein design. This adaptability and extensibility make it a platform that evolves alongside advancements in computational biology and protein science.

5.4.3 Benefits of PS-GO and SilverSurfer

PS-GO's development represents a significant advancement in the field of protein search engines, providing a robust and efficient platform for navigating extensive protein databases. Its advanced search capabilities allow for high-precision queries, which are essential for identifying proteins with specific functional characteristics. This precision enhances the ability of researchers to efficiently utilise vast datasets, ensuring that they can quickly find and analyse proteins that meet their specific criteria. The integration of multiple search parameters allows for a multi-dimensional approach to protein research. This capability saves valuable research time and enhances the depth and breadth of protein studies.

SilverSurfer excels in the tailored generation of proteins, using its customised genetic algorithm to explore and create novel protein sequences that meet precise functional requirements. This ability to generate proteins with desired properties through an evolutionary-inspired method allows for the design of proteins that may not naturally exist, offering potential solutions for unique therapeutic and industrial applications. The scalability and extensibility of SilverSurfer are also key benefits, as the platform can handle a growing amount of data and complexity, making it adaptable to future advancements in protein science. Its design supports seamless integration with other computational tools and platforms, ensuring that SilverSurfer can continually evolve and incorporate new research findings and technologies.

5.4.4 Synergy Between PS-GO and SilverSurfer

The collaboration between PS-GO and SilverSurfer introduces an innovative synergy that leverages the unique capabilities of each system to enhance the field of parametric protein design. This is exemplified through their integrated functionality, enhanced protein design capabilities, and streamlined workflow that collectively advance both the exploration and application of novel protein designs.

The integration of SilverSurfer as the sequence generator backend for the PS-GO system exemplifies a strategic joining of capabilities, creating a cohesive system that harnesses the strengths of both tools. This integration ensures that users can engage with a seamless interface that combines the deep search functionality of PS-GO with the generative abilities of SilverSurfer. By aligning the data handling and processing infrastructure of PS-GO with the genetic algorithm-driven design capabilities of SilverSurfer, the combined platform facilitates a dynamic and comprehensive approach to protein design.

Through the integrated system, researchers can directly utilise SilverSurfer's generative abilities within the PS-GO interface, allowing for the exploration and realisation of novel protein designs that are both innovative and tailored to specific scientific needs. This integration facilitates an environment where theoretical protein models can be rapidly prototyped and iteratively refined, accelerating the design cycle. Researchers can define desired protein characteristics in PS-GO and immediately leverage SilverSurfer to generate sequences that meet these parameters, effectively bridging the gap between search and synthesis. This capability creates the potential for breakthroughs in protein engineering, particularly in developing therapeutic proteins with optimal functional properties.

The synergy between PS-GO and SilverSurfer streamlines the workflow for protein researchers, combining robust search and comparison features with powerful generative algorithms. This streamlined approach enhances both the discovery and design process, enabling researchers to swiftly move from identifying potential protein candidates through PS-GO's extensive database and analytical tools to actively designing and generating novel proteins with SilverSurfer. The cohesive workflow reduces the operational overhead associated with switching between different platforms and tools.

5.5 Conclusion

This chapter has extensively explored the realm of parametric protein design, highlighting the pivotal roles of SilverSurfer and PS-GO in advancing this field. Through the application of innovative computational models and algorithms, they significantly enhance the efficiency and precision of protein design and discovery processes.

SilverSurfer, with its specialised genetic algorithm, has effectively showcased the ability to navigate and explore the vast protein sequence space, generating novel

protein sequences that meet specific functional criteria. This capacity to generate tailored proteins supports the development of unique therapeutic and industrial applications, underscoring the potential of genetic algorithms in transcending traditional boundaries in protein engineering.

PS-GO has emerged as a SOTA parametric search engine that excels at handling and analysing extensive protein databases with remarkable precision and speed. By integrating advanced data management and search techniques, PS-GO enables researchers to perform highly targeted searches, thereby significantly reducing the time and effort required to identify proteins with desired characteristics. This optimisation of the search process contributes to a more efficient allocation of resources in research and development endeavours.

The synergy between PS-GO and SilverSurfer further amplifies their individual strengths, providing a unified point of access that facilitates both the discovery and design of proteins. This collaborative platform accelerates the interactive cycle of protein design and testing.

The innovations presented by SilverSurfer and PS-GO represent significant advancements in the field of parametric protein design. Their contributions are poised to have lasting impacts on the biotechnological and pharmaceutical industries, where there is a demand for faster and more accurate methods of protein design.

Chapter 6

Protein Platform, Profasa & ChemFlow

This chapter addresses the integration of computational tools into practical workflows, focusing on user experience and accessibility. It follows the development of PROFASA [199] and ChemFlow [193] platforms. It covers the design and implementation of these systems, their impact on protein design and analysis, and the innovations in bioinformatics workflow automation.

6.1 Introduction

The understanding of protein structure and function constitutes a cornerstone of modern life sciences. Advances in computational biology have enabled us to decipher the molecular mechanism of proteins and gain the ability to tackle highly significant and difficult problems [161] such as predicting the 3D conformation of a protein and even designing novel protein structures *in silico*. This capability marks a major leap forward, extending our reach from mere observation to the proactive design of proteins with tailored functionalities that never existed in nature previously.

In recent years, the field of protein research has increasingly incorporated *in silico* experiments [49, 187], highlighting the shift towards computational methods as integral components of scientific inquiry. Computational tools such as AlphaFold have demonstrated remarkable success in accurately predicting protein structures based on amino acid sequences [143]. These advancements underscore a transformative period in protein science, where the ability to simulate and visualise complex protein interactions and conformations accelerates both fundamental research and applied biological engineering.

Scientific advancements are increasingly supported and enabled by the complex interactions between even more complex software systems [49, 187]. The integration of experiment and computational sciences is becoming crucial for cutting-edge discoveries in protein function and structure. A robust array of freely available protein tools supports these endeavours, spanning sequence alignment offered by Blast [11], structure prediction through Rosetta [173], I-TASSER [324], and the groundbreaking AlphaFold [143]. Visualisation tools such as ChimearX [231], PyMOL [330], and

CSynth [296] enrich the scientific community's ability to interact with and learn from protein models, enhancing the overall understanding of protein dynamics.

Despite the plethora of available tools and their profound capabilities, the application of SOTA methods in protein research remains challenging. The rapid evolution of computational methods often translates into complex, resource-intensive software that is not user-friendly. Tools developed primarily in high-performance languages such as C or C++ prioritise computational efficiency but often lack accessibility. These programs may be cumbersome to install, have steep learning curves, and require significant computational resources, which can deter widespread usage among researchers without extensive computational backgrounds. The challenges of employing the SOTA methods are further enhanced by the high rate of development of new models [187].

For example, AlphaFold, while revolutionary, demands sophisticated hardware setups and advanced programming skills to operate effectively. Similarly, I-TASSERS's computational processes are so resource-intensive that their computation is measured in days and is not feasible to run on conventional PCs. Furthermore, tools like RpsBlast, although powerful, feature complex command-line interfaces that pose substantial barriers to those unfamiliar with such environments.

As the scientific community continues to advance, the frontiers of protein science, enhancing the usability and integration of computational tools, will be essential. The goal moving forward is not only to maintain the computational power and precision offered by these tools but also to improve their accessibility and ease of use. This chapter will explore the development of protein design platforms, focusing on their critical roles in navigating the vast and dynamic landscape of protein science.

6.1.1 The Value of Workflows

Workflows are fundamental to the scientific process, serving as the structured methodology by which experiments are designed, executed, and evaluated. In the context of computational biology and protein design, the importance of clearly defined workflows cannot be overstated. These systematic protocols are crucial for ensuring that experiments can be replicated and the results can be validated, which is one of the core principles of scientific inquiry.

Describing workflows in detail is essential for any experiment. The ability to replicate experiments and achieve the same results under the same experimental conditions is foundational to the scientific method. Without detailed workflows, the reproducibility of scientific findings would be compromised, hindering peer verification and the advancements of knowledge.

Workflows bring structure, efficiency, and clarity to the research process. They

delineate the steps involved in an experiment, detailing the required inputs, the expected outputs, and the methods for achieving these results. This structured approach minimises errors and inefficiencies by standardising procedures and ensuring that all necessary steps are followed.

Well-defined workflows facilitate collaboration among scientists. In the increasingly interdisciplinary field of protein design, where experts from various domains, such as bioinformatics, molecular biology, and computer science, must work together, having a common framework and language is indispensable. Workflows enable team members to understand each other's roles and contributions clearly, ensuring that the integrated efforts are aligned and coherent.

6.1.2 Motivation

The motivation for developing advanced protein design platforms is undermined by the pursuit of enhancing drug discovery and supporting industrial applications. As the field of bioinformatics expands, the technical barriers associated with using SOTA models pose significant challenges. By lowering these technical costs, protein design platforms can empower scientists to concentrate more on the domain-specific aspects of their work, such as experimental design and hypothesis testing, rather than on the complexities of software management and integration.

One of the core motivations in the development of these platforms is the creation of a unified interface that serves as a comprehensive resource for all protein-related needs. This "one-stop-shop" approach is designed to streamline the protein design process, making it more efficient and user-friendly. By integrating various tools and functionalities into a single platform, researchers can access a wide array of resources in one place, from sequence analysis and structural prediction to protein modification and visualisation. This integration saves valuable time and enhances the effectiveness of protein design projects by providing a powerful toolkit that supports the rapid iteration and refinement of models and hypotheses.

The push towards adopting an "orchestrator" design in protein platforms is motivated by the need to make these tools more accessible and intuitive. Modern platforms increasingly incorporate NLP capabilities to allow users to input commands and define parameters in natural language. This shift towards human language as a mode of interaction reduces the learning curve and makes the sophisticated tools more accessible to a broader range of scientists, including those without extensive computational backgrounds. By simplifying the user interface and communication methods, these platforms can bridge the gap between complex computational processes and the everyday language of scientific inquiry, thus broadening the user base and facilitating the adoption of SOTA models.

These are some of the core reasons that collectively drive the development of next-generation protein design platforms, aiming to transform the landscape of pharmaceutical development and molecular engineering through enhanced accessibility, efficiency, and user engagement.

6.1.3 Contributions

This chapter highlights the development of a single point-of-contact platform that integrates a comprehensive suite of tools into a user-friendly interface suitable for both professionals and amateurs. This unified platform simplifies the user experience by consolidating multiple functionalities, allowing researchers to focus more on scientific inquiry and less on navigating complex software systems.

Another major contribution is the capability to generate detailed scientific workflows based on natural language descriptions. Utilising advanced natural language processing technologies, the platform translates user descriptions into executable workflows. This feature makes sophisticated computational tools accessible through everyday language and introduces innovative, prompt engineering techniques to ensure accuracy and alignment with scientific goals.

The platform also serves as a proof of concept for processing workflows. This is achieved through a novel pipeline for protein design that automates various computational tasks, allowing users to concentrate on the conceptual rather than the technical aspects of protein design.

Lastly, the chapter presents the development of optimised language model prompts for protein design, improving the platform's ability to process complex scientific queries with greater precision. These contributions collectively enhance the accessibility and efficiency of protein design tools, paving the way for more robust scientific discovery and innovation in the field.

6.2 PROFASA: Enhancing Protein Analysis and Design

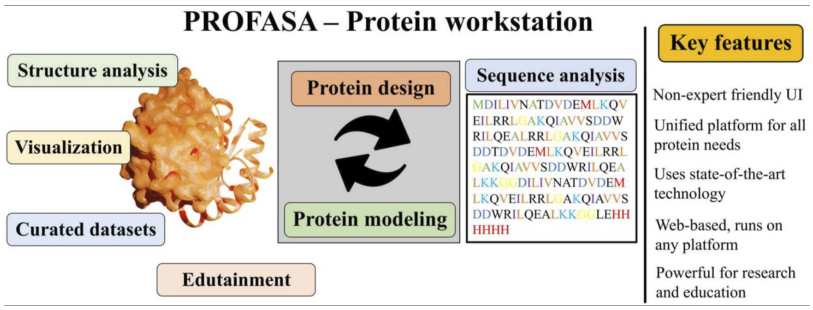


FIGURE 6.1: PROFASA combines all the major categories of tools, providing a single platform interface for all computational protein works. Source: Mi et al. [199].

PROFASA (Protein Fragment And Structure Analysis) represents a significant leap forward in the landscape of computational tools for protein analysis. It embodies a comprehensive *in silico* workstation designed to streamline and enhance the assessment of protein structures, parametric analysis, and visualisation. The platform consolidates a variety of sophisticated functionalities into a single, user-friendly interface that demystifies the complexities traditionally associated with protein research.

One of the core values of PROFASA lies in its ability to provide rapid domain analysis of input sequences, complemented by advanced molecular visualisation capabilities. This integration enables users to identify and visually explore the structural motifs within proteins, fostering a deeper understanding of the link between structure and function. The platform supports the construction of protein models with the click of a button, utilising renowned tools like I-TASSER [324] and AlphaFold [143]. This feature allows for the simultaneous folding of multiple input sequences, generation of novel protein sequences, and modelling, accelerating the design process.

By simplifying the submission process, where users need only input a sequence to receive a comprehensive analysis, including parameters like accessibility, instability, hydrophobicity, isoelectric point, and Ramachandran plot scores, PROFASA ensures a seamless user experience. The platform's robust capability to provide accurate, reliable results across all its features further enhances its utility in the scientific community.

Data visualisation and comparison are handled by utilising tables and plots to represent data in an intelligible manner. This not only aids in the interpretation of results but also in comparing different protein models. The F2F (function to form) [323] serves as a pivotal tool within PROFASA, guiding users through the expected performance of proteins based on their structural and functional attributes.

PROFASA incorporates sequence alignment enabled by the NCBI BLAST toolkit and structure prediction managed through I-TASSER [324] and AlphaFold [143]. Other essential protein parameters are calculated using scientifically validated formulas, ensuring the outputs are accurate and credible.

The user interface of PROFASA is a testament to its design philosophy, offering an intuitive experience that caters to both seasoned professionals and those new to protein analysis. Its appealing interface facilitates a deeper engagement with the tool, making advanced protein design and analysis accessible to a broader audience, thereby democratising the capabilities of computational protein research.

6.2.1 Architecture Design

The architecture of PROFASA is designed to function as a comprehensive one-stop workstation for protein sequence and structure-based analysis. This platform is crafted to meet the diverse needs of users, ranging from academic researchers to industry professionals, by providing an all-encompassing suite of tools.

PROFASA presents a classical Model-View-Controller design pattern commonly used in the software development of enterprise applications, allowing each layer to be tailored based on specific roles within the system. The architecture is segmented into three primary layers, each dedicated to a specific aspect of the platform's functionality. The front end, or the View component, includes various user interface pages such as the Home Page, Blast Page, Fold Page, Resource Page, Results List Page, Text Result Page, and 3D Result Page. These pages are constructed according

to the tenets of UX/UI (User Experience/User Interface), facilitating easy navigation and efficient access to the platform's extensive features.

The Logical layer, or the Controller in the MVC pattern, is managed via Node.js. This part of the backend handles dynamic aspects of PROFASA, including account services, mail services, note services, sequence services, and specific services for I-TASSER and AlphaFold. This layer ensures robust data handling and seamless integration of services, facilitating a responsive and reliable analytical environment.

The Computing layer, representing the Model in the MVC framework, is where the computational heavy lifting occurs. It is implemented through a command-line program interface, incorporating tools such as RpsBlast, I-TASSER, AlphaFold and Ramachandran. This setup allows PROFASA to efficiently handle computationally intensive tasks, providing users with fast and accurate analysis results.

Front End

The front end of PROFASA is constructed to cater to a diverse user base, ranging from seasoned professionals in the field of protein research to non-experts interested in exploring protein structures and functions. This inclusive approach is reflected in the interface design, which emphasises visual clarity, user-friendly interaction modes, and minimised learning costs to accommodate users with varying levels of technical expertise.

The main programming language utilised in the development of the front end is TypeScript, a superset of Javascript that incorporates additional useful features such as static typing, interfaces, and classes [25] attempting to enhance code quality and readability. TypeScript's robust typing system and object-oriented capabilities make it a top contender for developing complex web applications such as PROFASA.

ReactJS, a popular JavaScript library for building user interfaces, is chosen as the development framework due to its declarative and component-based architecture. This framework facilitates the efficient rendering of interactive UIs that respond dynamically to user inputs and data changes, making it particularly suitable for the complex data involved in protein analysis platforms.

To manage application routing and state, PROFASA employs UmiJS, an enterprise-level React application framework. It enhances the application's architectural capabilities by providing built-in routing functions and a comprehensive state management system based on React hooks. This framework supports modular and maintainable code development, which is crucial for handling the workflows of protein analysis.

Visual rendering and interaction with protein 3DS models are primarily handled using Three.js and ngl.js. Three.js, established by Ricardo Cabello [67], facilitates the creation of WebGL-based 3D graphics. It is extensively utilised within PROFASA for rendering protein structures with detailed lighting, camera controls, animations and interactive features such as mouse interactions. Ngl.js specialises in

loading and parsing PDB files and is pivotal in tasks such as superimposing protein sequences and modifying the visual representation of proteins by altering the colours and shape of selected fragments.

For graphical representations such as radar plots, which are used to visualise and compare the F2F results of proteins, the front end utilises the HTML5 Canvas element. This hand-crafted implementation approach is motivated by the need for extensive flexibility, and to ensure consistency between the displayed visuals on the web interface and those downloaded by the user.

The front end includes a document download feature implemented with the React PDF Renderer library. This feature enables users to copy individual protein sequences, download PDB files, download diagrams, or even download comprehensive PDF documents containing detailed protein analyses. This feature enables easy documentation and facilitates sharing of protein data.

Back End

The back end of PROFASA is developed using JavaScript and operates within the Node.js environment. Node.js serves as a runtime environment that leverages Google's Chrome V8 JavaScript engine, functioning outside the browser. This allows for efficient management of asynchronous events and supports high-performance server-side operations essential for robust web applications.

For the web server framework, PROFASA utilises KOA, a powerful middleware system designed around Node.js, adept at handling a variety of tasks such as request processing, Cross-Origin resource sharing (CORS) processing, token authentication, request interception, error handling, and logging.

The database interactions are managed using Sequelize, an Object-Relational Mapping (ORM) framework that supports robust interaction with SQL databases without the necessity of writing SQL code. Sequelize greatly simplifies database transactions, data retrieval, and relationship management between code, data structures and database tables. It also facilitates database migration and the setup of test environments, enhancing the back end's scalability and maintainability.

Node.js plays a central role in the PROFASA architecture, as it establishes servers, listens to requests, and handles the read and write operations for databases. Additionally, Node.js is utilised to run a variety of command-line scripts, which are crucial for performing the complicated functions necessary for protein analysis. These functions include the execution of open-source third-party programs and custom Python scripts. To execute these command-line programs, which are predominantly written in C and C++, PROFASA employs the Node.js standard library's child process.exec() and child process.execSync() methods. This approach allows for synchronous and asynchronous command execution directly from the main Node.js program, integrating the complex computational capabilities into the web platform.

Profasa acts as a cohesive framework that integrates well-known protein APIs, such as RCSB search and Protparam. This integration facilitates direct calls to these

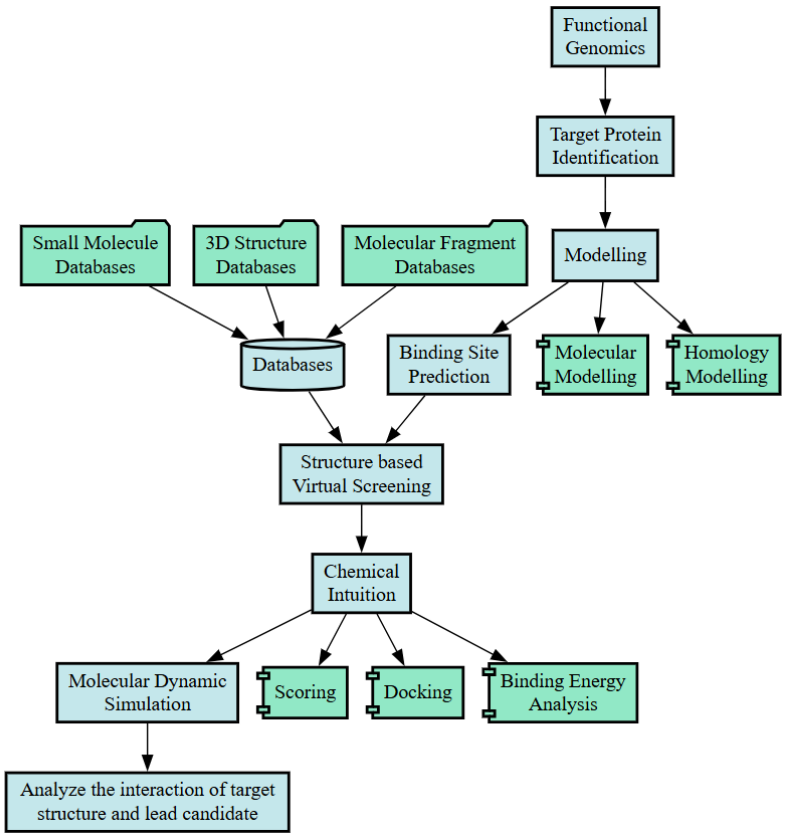


FIGURE 6.2: Workflow example for a typical computational drug development process.

services, allowing PROFASA to use external computational resources and databases, expanding the protein analysis capabilities. By acting as the glue that binds together diverse functionality and external resources, PROFASA delivers a comprehensive and versatile platform for protein analysis and design.

6.3 ChemFlow: Automating Bioinformatics Workflows

ChemFlow is a novel platform that revolutionises how scientific workflows are conceptualised and executed within the field of bioinformatics. It serves as a proof of concept, uniquely translating natural language descriptions directly into executable scientific workflows through advanced prompt engineering techniques and executing them. This platform eliminates the need for traditional coding, thereby reducing the skills barrier associated with running SOTA models and reducing the potential security risks associated with code execution. ChemFlow is tailored for protein design, enabling researchers to specify, modify, and execute complex bioinformatics workflows with unprecedented ease and accuracy, streamlining the path from concept to execution in protein research.

6.3.1 Workflows

ChemFlow harnesses the power of scientific workflows to streamline and optimise the execution of complex bioinformatics processes. Scientific workflows are structured, repeatable procedures that map out a sequence of tasks aimed at achieving a specific scientific goal (Fig. 6.2). These workflows are critical in managing the complexities of data and process dependencies, especially in fields like drug design, where precision and reproducibility are paramount.

A typical scientific workflow in ChemFlow is composed of a series of well-defined actions, each with specific input and output data, directed towards a defined objective. These activities are often visualised as a directed acyclic graph (6.2), where each node represents a task and edges define the dependencies between these tasks. This graphical representation ensures that dependencies are respected and that tasks are executed in the correct sequence, enhancing the clarity of the process.

The use of workflows in ChemFlow is primarily aimed at accelerating scientific discovery. By automating repetitive tasks, the platform reduces human error and frees researchers to focus on more complex analytical tasks. Workflows are designed to scale efficiently with increasing data sizes or computational needs, which is essential for managing the expanding datasets typical in bioinformatics. Furthermore, once established, these workflows can be easily adapted and reused for different projects with similar requirements, significantly reducing the startup time and effort for new initiatives.

Workflows may range from simple, linear chains of tasks to complex graph-structured configurations, depending on the complexity of the objectives and data involved. Typically, a workflow includes the input, which could be raw or pre-processed data such as genetic sequences, protein structures, or drug compound databases; the transformations, which are computational or analytical operations applied to the input; and the output, which are the final results such as newly designed drug compounds, a set of protein structures, or detailed report of the findings.

6.3.2 LLM

LLMs, also known as foundational models, are transformative machine-learning technologies that have the potential to reshape human-machine interaction. These models, such as ChatGPT from OpenAI [39], LLaMA from Meta [297], and PaLM from Google [57], are built on extensive transformer-based architecture. Due to the massive amounts of internet-scale data employed in their training, they gain general-purpose language processing and generation capabilities, such as text summarisation [40], generating creative content [40], question-answering [40] and even problem solving [40, 326].

The quest to bridge the communication gap between humans and machines has evolved significantly over time. From initial efforts to encode voltages in wires as binary digits to the creation of high-level programming languages with rigorously

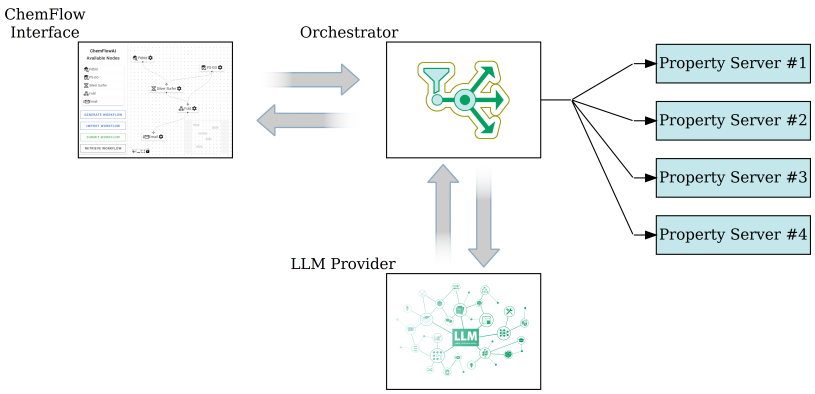


FIGURE 6.3: ChemFlow Architecture.

defined syntaxes, each advancement has brought the point of communication closer to natural language. LLMs represent a critical milestone in this evolution, establishing a foundation for a future where the symbiotic relationship between researchers and machines is deeply enriched.

In the context of ChemFlow, LLMs play an essential role in lowering the learning curve and democratising access through the translation of natural language descriptions into workflow representations. These models interpret the textual descriptions provided by users to delineate the workflow's structure, detailing the nodes, their attributes and parameters, and defining the connections between them. This capability enhances the platform's usability by allowing researchers to input complex workflow descriptions in a simple, intuitive language, which the LLM then converts into a structured format ready for computational processing.

6.3.3 Architecture Design

One of the main design objectives of ChemFlow is to streamline the complex problem of protein design, with an accent on user accessibility. At its core, ChemFlow seeks to alleviate the technical burdens often encountered by scientists, enabling them to engage with the platform through intuitive and natural language interfaces. This user-centric approach ensures that the platform is not only powerful but also accessible to researchers from diverse backgrounds, regardless of their computational expertise.

ChemFlow's architecture is partitioned into distinct layers, each serving a different purpose in the workflow of protein design. The presentation layer, or the front end, is crafted to provide an interactive and intuitive user experience, facilitating effortless expression of intent. This interface is the gateway through which users interact with the system, and for this reason, it prioritises a user experience that is intuitive for beginners yet robust enough for advanced users.

The logical layer acts as the orchestrator, seamlessly integrating the various components of ChemFlow. This middle layer coordinates the interaction between the user interface and the computational backbone, ensuring the user workflows are

processed in the correct order and providing access to input data. It plays an essential part in being responsible for the workflow's data management and correct execution.

The computing layer, including the LLM provider and various Protein Property Clients, is responsible for the computational 'heavy-lifting', processing complex data and executing sophisticated algorithms that underpin the protein design process. The LLM provider, powered by OpenAI GPT [221] API, leverages large language models to interpret and transform natural language into configuration strings, while the Protein Property Clients handle specific tasks such as protein folding predictions and protein function predictions.

ChemFlow capitalises on the automation capabilities and extensive knowledge base of the LLMs while negating issues like hallucinations through the rigorous grounding and validation processes. This increases the reliability and accuracy of the outputs, supporting the intuitive interaction with the platform through natural descriptions.

Together, these layers form a robust architecture that supports the dynamic needs of protein design and enhances the research efforts through the easy application of scientific workflows. The integration of advanced computational techniques allows ChemFlow to offer significant advantages over traditional scripting approaches. Besides eliminating the need for programming skills, it also maintains a comprehensive record of the computational graph and parameters used, provides uniform access to computational components, and effectively manages the complexities associated with multiple compute nodes and job scheduling systems.

Front End

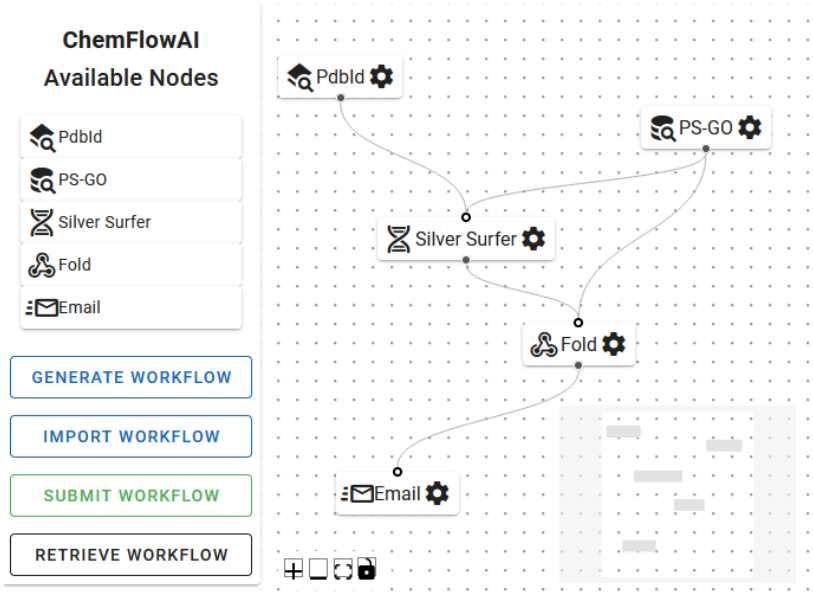


FIGURE 6.4: Snapshot of the Workflow Design GUI of ChemFlow.

The front end of ChemFlow (Fig. 6.4) serves as the user's primary point of interaction with the platform, offering an intuitive environment for designing and managing protein design workflows. Developed using the JavasScript programming language, ChemFlow's front end leverages the Vue.js framework along with the Vuetify component library to ensure a responsive and aesthetically pleasing user experience. This combination allows for a single-page web application that efficiently handles user interactions and data updates without requiring page reloads.

Within this framework, ChemFlow incorporates features that streamline the workflow creation and modification processes. Users have the unique capability to generate workflows from natural language descriptors through the platform's seamless integration with LLMs. For those who prefer a more traditional hands-on approach, an interactive drag-and-drop interface allows users to visually design workflows, placing and connecting nodes on a canvas in a manner inspired by visual programming paradigms.

The front end also provides a high-level visual representation of the LLM-generated workflows for detailed inspection and correction. This feature ensures the accuracy of workflow interpretation and enables straightforward modification before committing to the workflow and starting to its execution. The inspection is realised using the same component as the traditional drag-and-drop design interface.

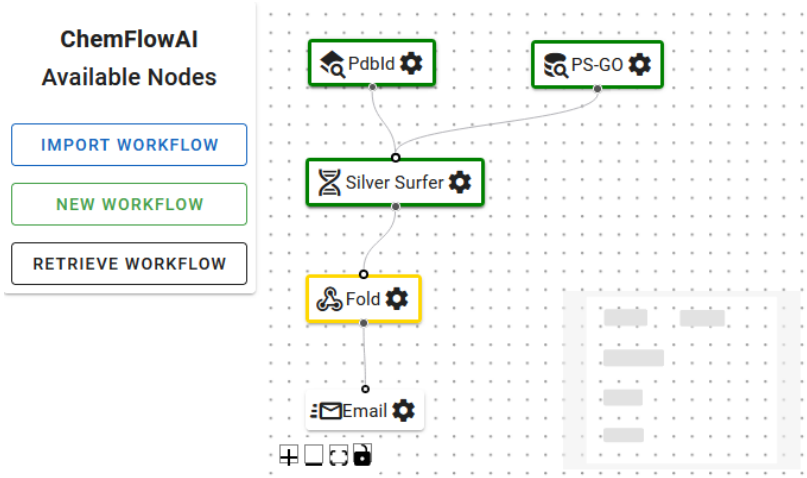


FIGURE 6.5: Snapshot of the Workflow Visualisation of ChemFlow. Green - Nodes that have finished processing; Yellow - Nodes currently being processed; White - Nodes yet to be processed.

Once the workflow is generated and the user has validated and submitted it for execution, a web-based dashboard displays the real-time status, allowing for monitoring of the progression of the computational tasks, in a component similar to the manual workflow design one (Fig. 6.5), designating nodes that have finished processing with green, nodes that are currently processing with yellow, and nodes yet to be processed without a coloured border. One significant distinction from the manual design view is that upon clicking on an executed node, the output, or a

summary of the output, can be visualised alongside the parameters used for the execution.

The front end of ChemFlow prioritises a user-centric design ethos, making advanced protein design tools accessible and functional for a diverse range of users, from expert researchers to those new to the field of bioinformatics.

Back End

The back end of ChemFlow is implemented using the Python 3.9.6 programming language. It leverages the Flask framework through the Flask RESTful API for request processing and handling, supporting the request-based communication channel between the user interface, the orchestrator, and a subset of the property servers.

The REDIS in-memory key-value store is essential to the back end architecture, serving as an in-memory data structure database that manages the state and data of the diverse workflows. This functionality is indispensable for maintaining the integrity and continuity of workflows, allowing for consistent performance across multiple sessions.

LLMs play a central role in ChemFlow workflow generation and, a significant support role in the execution of workflows. Initially, LLMs generate a comprehensive computational graph from the natural language description written by the user. The computational graph outlines the sequence of operations and their dependencies. After graph generation, LLMs configure each node within the graphs with the appropriate parameters.

The process of generating workflows involves several phases. Initially, the user-provided description, enriched with contextual information, is processed by an LLM to extract the computational graph. This graph details the necessary operations and their connection based on available plugin functionalities. Once generated, the computational graph undergoes a validation process to identify and rectify any errors, followed by an optimisation step to ensure the most efficient representation. Each node within the graph is then configured individually using the YAML API description to generate each node's settings.

The execution process is initiated upon a workflow submission, starting parallel processing for all nodes with no dependencies. As each node completes its tasks, the system updates its status and places the generated data on a dedicated data bus, which manages the flow and integrity of data across the workflow. Upon the placement of the data on the common data bus, a search is initiated for all the nodes that have all their dependencies resolved; each of the nodes with complete dependencies is triggered and executed. This process is repeated until all of the nodes have been processed.

The option to deploy the back end locally, and therefore, the data bus, brings a significant advantage in ensuring data integrity and security. The local deployment ensures that all sensitive information is securely maintained within company

infrastructure, safeguarding intellectual property and data privacy. The data bus integrates all provided and generated data throughout the execution process, ensuring data continuity and maintaining data governance.

ChemFlow's back end implements a novel error-handling mechanism through its automatic error resolution approach, which is enabled by the use of LLMs. The error text message, together with the YAML configuration files, are provided to the LLM in order to produce a solution and reduce the hallucination risk by grounding the LLM's output.

ChemFlow interfaces with a distinct array of external property computing servers, each dedicated to calculating specific protein properties. These servers operate outside the core ChemFlow infrastructure and communicate with the ChemFlow backend through web requests defined within their respective plugins. The current lineup includes servers such as PsGO, Silver Surfer (Chapter 5), PROFASA (Chapter 6) and Gmail, supporting functionalities like PDBID search for retrieving proteins based on their database identifiers, Parametric Search for retrieving proteins based on their property parameters, Silver Surfer for parametric protein design, protein folding for structure prediction, and emailing results for notification and distributing outcomes.

By integrating these advanced technologies, ChemFlow's back end supports the operational requirements of complex bioinformatics workflows and enhances the platform's overall reliability and user experience. Due to the plugin design of the property computing nodes, ChemFlow is poised to remain a cutting-edge tool in the field of protein design, capable of meeting the diverse needs of the scientific community.

6.4 Impact On Protein Design

The advent of ChemFlow and PROFASA into the protein design process represents a transformative approach that significantly accelerates scientific research by providing a unified point of contact for accessing and analysing protein properties. The consolidation into a single platform dramatically reduces the time and effort required for researchers to switch contexts between different computational tools and data endpoints. Traditionally, scientists have had to navigate through multiple systems to gather necessary data and perform diverse analyses, a process that is not only time-consuming but also prone to errors and difficulties during data transfer. By serving as a central hub, ChemFlow eliminates the need for constant switching and data transfer between different tools, thereby enhancing productivity and reducing the risk of data loss.

The automation of workflow generation and execution by ChemFlow significantly reduces the technical burdens traditionally placed on scientists. In the fast-evolving field of protein design, the ability to automate and integrate new SOTA models into complex workflows is invaluable. This automation supports researchers

by managing the technical complexities of data processing and analysis, allowing them to focus on scientific questions rather than the operational details of computational tools. ChemFlow leverages language models to interpret natural language and generate precise scientific workflows, simplifying the user experience and making advanced computational tools accessible to a broader scientific community.

Together, ChemFlow and PROFASA streamline the scientific process and enhance the reliability and efficiency of protein design projects. By automating and integrating essential tools and functionalities, these platforms provide researchers with powerful resources that support rapid iteration and refinement of hypotheses. This, in turn, speeds up the discovery process and ensures that the results are reproducible, pushing the boundaries of protein design and achieving a new standard for computational support in bioinformatics research.

6.4.1 PROFASA Results

The PROFASA platform is instrumental in transforming the landscape of protein design through its comprehensive suite of tools, which integrate analysis, superimposition, structure-based searching, and protein folding functionalities into a user-friendly interface.

PROFASA's Analysis section is designed to provide a deep understanding of various structural, sequence-based and functional parameters. Users initiate this process by uploading a PDB file, after which PROFASA computes various structural and functional metrics. The platform handles this data and produces a detailed PDF report containing the results.

The Superimpose function of PROFASA allows for the comparative analysis of protein structures. By overlaying two or more 3D structures, researchers can visually assess and study structural variations. This capability is crucial for examining the effects of mutations, alterations in functional sites, or the optimisation of novel protein fragments. PROFASA provides a live, interactive 3D visualisation of the input PDB files superimposed based on metrics such as the RMSD and GDT_TS.

The Structure Blast function unique to PROFASA extends beyond traditional sequence-based BLAST analyses by not only finding matches for full or partial user-defined sequences but also visualising these matches in 3D. This functionality enables users to see how domains and fragments from various smaller proteins might fit together in larger complex proteins, providing insights into protein evolution and function that are not readily obtainable through conventional methods.

For sequences with unknown structures, PROFASA can utilise tools from the Fold section to predict their conformations. This section has a focus on user experience, allowing users to fold multiple protein sequences simultaneously and offering them a choice between modelling tools like I-TASSER and AlphaFold. The generated structures can then be visualised in PROFASA's integrated, interactive molecular visualisation tool. Unlike other protein analysis and folding software, PROFASA's

one-stop-shop approach allows users to interact with their 3D protein structures directly on the same platform. This integration enhances user experience, providing an uninterrupted workflow and enabling more dynamic interaction with the protein models.

The platform's comprehensive toolset and integrated approach reduce the technical barriers typically associated with advanced protein analysis and design, as well as eliminating the context-switching cost, making it more accessible to a broader range of researchers.

6.4.2 ChemFlow Response

The assessment of ChemFlow's impact on protein design was evaluated through a combination of internal testing, user feedback, and small-scale experimental tests. These assessments have provided valuable insights into both the performance of the platform and its user interface, guiding a future development path.

Frontend

From the front end perspective, user feedback has been overwhelmingly positive, particularly with regard to the intuitive nature of the drag-and-drop interface. Users have appreciated the ability to construct and modify workflows visually, which significantly simplifies the process of designing complex bioinformatics experiments. The real-time visualisation of workflows has been highlighted as a desirable feature, providing immediate feedback on workflow structure and status.

Users have also expressed a desire for additional features, such as the implementation of workflow loops and automated triggers. These suggestions indicate a desire for more dynamic and flexible workflow management tools, automating repetitive tasks and responding to conditional changes. This feedback indicates possible paths for further development.

Backend

The backend of ChemFlow has demonstrated robust performance, effectively managing diverse workloads and ensuring reliable workflow execution. The architecture's ability to handle everything from simple tasks to more complex multi-step processes without imposing a significant system burden. This efficiency is largely attributed to the system's focus on processing API responses rather than heavy computational tasks.

The dual deployment capability, allowing users to operate the system either in a cloud environment or locally, has been particularly well-received. This flexibility not only caters to different user preferences and requirements but also enhances data security and operational control, which are crucial in sensitive research environments.

Workflow Generation

The Workflow Generation feature of ChemFlow plays a foundational role in streamlining the protein design process by automating the conversion of scientific intentions into executable workflows and in the automated error resolution. This functionality significantly reduces manual intervention, allowing researchers to focus on core scientific inquiries. Large-scale testing was not pursued due to the high cost and the challenges of creating the extensive datasets required to accurately sample the space of workflows. Instead, small-scale evaluations have been conducted to validate the approach.

In these small-scale tests, the generated workflows were found to be correct and optimal in 90% of cases, showing the high efficiency and accuracy of the approach. Out of the remaining 10%, 8% of the cases were found to produce the correct output but included unnecessary steps, and the rest of 2% of the cases failed to produce the desired output. These tests have been instrumental in the development of ChemFlow's prompting strategy.

User feedback has been overwhelmingly positive regarding the ability to generate workflows from natural language descriptions. Users appreciated this feature as it lowered the technological barrier to entry for complex bioinformatics tasks. The translation from natural language to executable workflows generally met user expectations and was seen as a valuable tool for enhancing user engagement and interaction.

Overall, the response to ChemFlow indicates a strong foundation with ample room for further enhancements. The feedback has been instrumental in identifying key areas for further development and ensuring the platform covers the needs of the scientific community, facilitating protein design.

6.5 Technical Innovations - LLM prompt optimisation

The optimisation of language model prompts represents a nascent field in machine learning. ChemFlow incorporates a novel set of prompt optimisation strategies for refining the interaction between human researchers and machine learning models. These strategies include prompt style transfer, prompt rephrasing, iterative improvement and ranking, and prompt correction. Each strategy is designed to improve the quality and accuracy of the responses generated by the large language model according to a different aspect of natural language understanding.

6.5.1 Prompt Style Transfer

The Prompt Style Transfer strategy (Fig. 6.6 a) is one of the four prompt optimisation pillars supporting ChemFlow. This technique involves instructing the LLM to reinterpret user descriptions by converting them into the technical language and style characteristic of computer science. The motivation behind this strategy stems from

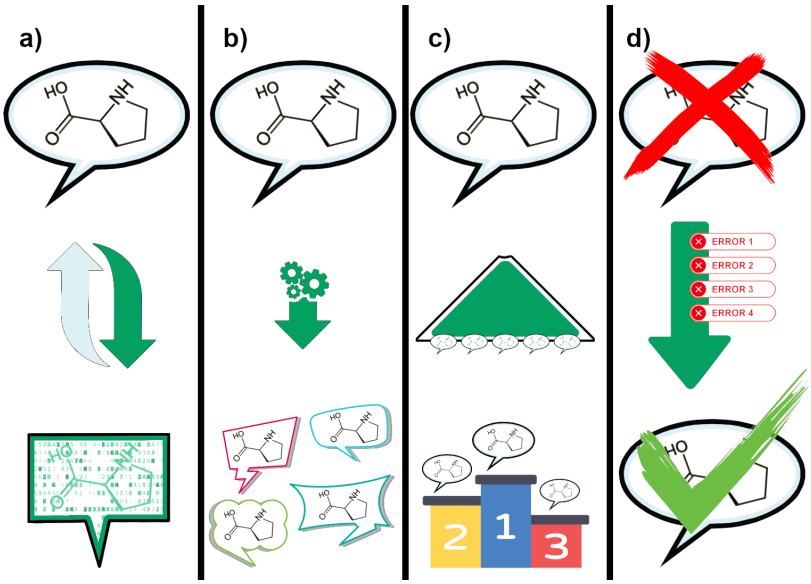


FIGURE 6.6: Visual representation of the 4 optimisation concepts employed by ChemFlow; a) Prompt Style Transfer, b) Prompt Rephrasing, c) Prompt Iterative Improvement and Ranking, d) Propmt Correcting.

the observation that LLMs generally perform sub-optimally in languages other than English.

Empirical experiments have demonstrated that utilising a well-defined technical language can significantly enhance the quality of the outputs generated. This improvement is believed to be due to the LLM's architecture, which functions akin to an ensemble of roles tailored to different contexts and styles of communication. By employing specific technical terminology, the model is prompted to adopt a more analytical and precise "technical role," facilitating more accurate and context-appropriate responses. This effect is supported by the findings of Bhargava et al. in their "What's the Magic Word? A Control Theory of LLM Prompting" [24] work. Further supporting this hypothesis is the work of Yin et al. [328]; they have observed that the effectiveness of the network is influenced not only by the language but also by the level of politeness used in the prompt.

6.5.2 Prompt Rephrasing

Prompt Rephrasing (Fig. 6.6 b) is another of the prompt optimisation pillars supporting ChemFlow; it focuses on enhancing the efficacy of language model interactions by generating alternative descriptions that retain the core essence of the initial prompt. This technique is rooted in the observation that subtle variants in the initial words of a prompt can significantly influence both the performance and the direction outcome of the model's responses. This phenomenon, where slight lexical shifts can steer the computational process in entirely different directions, is supported by the findings of Bhargava et al. [24].

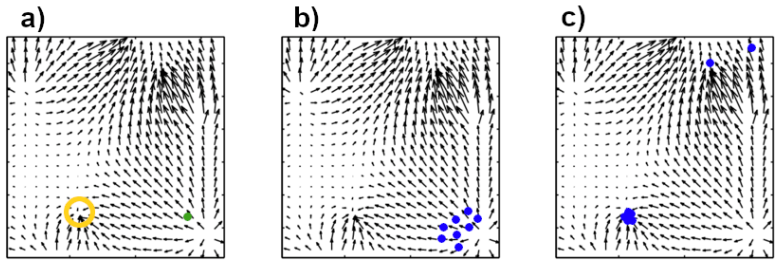


FIGURE 6.7: Visual depiction of the conceptual inspiration for the Prompt Rephrasing strategy. a) Yellow circle denotes the correct answer; Green dot denotes the starting point with the initial prompt; b) Blue dots indicate starting points obtained from prompt rephrasing; c) Blue dots after finishing the prompt evaluation.

The methodological inspiration for prompt Rephrasing comes from Genetic Algorithms. In this context, we treat each rephrased prompt as a member of a population, where the generation of the next token is analogous to following a high-dimensional flow field described by the LLM's inherent nature. This analogy helps us appreciate the potential of starting with multiple variations of a prompt: just as generic diversity in biological populations increases the chances of reaching an optimal solution, introducing linguistic diversity in prompt formulations encourages the model to follow different paths in the same flow field, escaping potential local optimum spots and increasing the chance of finding optimal solutions.

Thus, by deploying multiple rephrased prompts that orbit around the same conceptual nucleus, we effectively explore a dense neighbourhood within the model's flow field. This strategy increases the likelihood of eliciting at least one highly accurate and contextually appropriate response from the LLM, thereby enhancing the overall reliability and precision of the tool in translating complex scientific queries into actionable computational workflows.

6.5.3 Prompt Iterative Improvement & Ranking

The strategy of Prompt Iterative Improvement & Ranking is a third pillar supporting ChemFlow. This strategy leverages the iterative refinement of responses and the ranking of multiple solutions to achieve high-quality and precise outcomes. The approach draws inspiration from concepts discussed by Bubeck et al. [40].

The iterative improvement aspect of the strategy focuses on the enhancement of existing solutions provided by the network. Initially, the LLM generates a baseline solution based on the user's initial prompt input. This solution may be further optimised. The iterative improvement process involves submitting this initial solution back to the LLM, accompanied by a request for more refinement. The model is tasked with reevaluating the solution, identifying potential inefficiencies and proposing modifications or alternatives to enhance the quality and performance of the outcome.

The ranking aspect of this strategy involves presenting the LLM with multiple generated solutions to the same problem. The model evaluates these alternatives, focusing on the solutions' fidelity to the original prompt and their computational efficiency. The ranking process prioritises solutions that adhere closely to the user's specified requirements and show a higher efficiency in the amount of computing resources used to execute the workflow.

Together, these aspects provide a mechanism for refining and selecting the optimal solutions in an LLM-driven workflow. Through iteratively improving responses and evaluating the responses, ChemFlow can help improve the quality of the produced responses.

6.5.4 Prompt Correcting

The Prompt Correction strategy represents the last prompt optimisation pillar that supports ChemFlow. This method involves presenting the LLM with a previous incorrect response alongside the error message generated during API calls or computational validation. The model is then tasked with analysing the feedback and generating a corrected answer. This approach is particularly effective for two main reasons.

Firstly, the training corpus of LLMs often includes substantial amounts of conversational data from technical forums such as Stack Overflow. When an error is reformulated in the context of a query, it mimics the format of questions typically found in these forums. This familiarity encourages the LLM to switch to a "problem-solving role", enhancing its ability to understand the context of the error and provide a more accurate solution.

Secondly, focusing on the error specifics helps pinpoint the exact nature of the problem, providing the LLM with essential insights into where the computational process deviated from expected outcomes. The error text not only highlights what went wrong but often also contains clues about how to correct the issue. By leveraging this detailed feedback, the model can adjust the erroneous solution to the correct one.

This strategy improves the accuracy of the results and significantly grounds the network's responses, ensuring that they are aligned with the computational constraints. Prompt Correcting offers ChemFlow an automated error resolution mechanism.

6.6 Challenges and Future Directions

6.6.1 Challenges

Despite the significant advancements and capabilities presented by ChemFlow and PROFASA, several challenges and limitations persist that could impact the effectiveness and scalability of these platforms. Understanding these challenges is fundamental for ongoing development.

One of the fundamental limitations faced by ChemFlow and PROFASA arises from their reliance on multiple external tools to perform various analyses. While this integration enhances the platforms' capabilities, it also introduces a dependency on the accuracy and reliability of these external tools. Any errors, limitations, or inaccuracies in the external tools will directly affect the final outcomes produced by ChemFlow and PROFASA. This dependency makes it imperative to continuously monitor and validate the external tools to ensure that they meet the required standards for accuracy and reliability.

Another of the challenges faced by the PROFASA platform is the computational intensiveness of the analyses performed. The process of folding proteins, predicting functions, and performing complex simulations requires substantial computational power and time. This intensity can lead to high operational costs, especially in terms of computational resources. Maintaining the necessary infrastructure can be expensive and challenging, particularly for facilities with limited budgets.

An additional challenge is presented by the rapid technological advancements and evolution of SOTA models in bioinformatics. To remain effective and relevant, platforms like ChemFlow and PROFASA must regularly integrate new and updated models into their systems. This ongoing need for integration requires constant updates to the platforms and may disrupt existing workflows. Furthermore, each integration demands testing and validation to ensure that the new models don't adversely affect existing workflows.

While ChemFlow and PROFASA employ advanced architectural designs and prompt engineering to mitigate security risks, one significant threat that is difficult to harden against is the engineering of malicious prompts. ChemFlow partially mitigates this by not executing any code generated by the LLM. Despite this, if there is a vulnerability in any of the parameters of any of the third-party services employed by the platforms, a malicious actor could potentially leverage prompt engineering in order to exploit that vulnerability.

Both ChemFlow and PROFASA offer powerful tools for protein design and analysis, yet they face several challenges that could impact their long-term viability. Addressing these challenges requires a proactive approach to platform maintenance, updates, and security management.

6.6.2 Future Directions

As we advance into the future, several enhancements and expansions are planned for both ChemFlow and PROFASA to further revolutionise the protein design landscape. These future directions are guided by the commitment to improving user experience, enhancing platform capabilities, and maintaining alignment with the latest scientific advancements.

To enhance the versatility of ChemFlow and PROFASA, additional computational tools and databases can be integrated into the platforms. This expansion can include more specialised tools for niche areas of protein search, such as membrane protein modelling. The integration of diverse tools will provide users with a more comprehensive suite of options for protein analysis and design, facilitating a deeper exploration of protein functionalities and interactions.

Recognising the recurring needs of many users, an integrated library of generic workflows can be integrated into ChemFlow. These generic workflows would address common tasks in protein design and would serve as a valuable resource, allowing users to either quickly deploy standardised workflows or to serve as a quick startup for personal workflows. These pre-configured workflows would be optimised for performance and validated for accuracy, ensuring that users can rely on them to produce reliable and reproducible results.

To increase the efficiency of data processing within ChemFlow, a further development avenue is constituted by the implementation of eager execution strategies. This approach would allow the platform to start processing dependent nodes on partial data without waiting for all the dependency nodes to complete. Eager execution would significantly reduce the turnaround time for experiments.

A further direction of development is presented by the potential of LLMs to extend beyond simple workflow generation. This avenue explores the possibility of using LLMs to assist in the deployment and configuration of Docker instances. This initiative would provide users with greater flexibility and control over their computational environments. It would also significantly widen the set of available tools supported by the platform. This capability would potentially allow automated SOTA model integration.

These future directions enhance the functionality of ChemFlow and PROFASA while ensuring that they remain at the forefront of technological advancements in protein design. By continuously integrating new features and capabilities, the platforms support the evolving needs of the scientific community and foster innovation in the field of bioinformatics.

6.7 Conclusion

This chapter has explored the transformative impact of integrated platforms like PROFASA and ChemFlow on the field of protein design and computational biology. These platforms embody the convergence of advanced computational methods

and user-centric design, offering tools that significantly enhance the effectiveness of protein analysis.

PROFASA and ChemFlow serve as exemplary models of how sophisticated computational capabilities can be made accessible to a broad range of users, from seasoned researchers to those without extensive experience. By automating complex workflows and lowering the barriers to entry, these platforms democratise high-level protein design and analysis, enabling more scientists to participate in and contribute to cutting-edge research.

The integration of LLMs within ChemFlow highlights a forward-looking approach to interface design, where natural language processing facilitates a more intuitive interaction between humans and machines. This innovation not only simplifies the user experience but also streamlines the translation of scientific intent into actionable computational tasks.

The chapter also acknowledges the challenges that accompany the development and operation of such sophisticated systems. The dependency on external tools and the need for constant updates and validation to ensure reliability and accuracy. Moreover, the computational intensity of some tasks and the rapid evolution of SOTA models present additional challenges to the platforms.

Looking ahead, the continued development and enhancement of PROFASA and ChemFlow are pointed to further revolutionise the landscape of protein design. Possible expansions include the integration of additional computational tools, the development of generic workflow libraries, and the exploration of eager execution strategies to enhance processing efficiency. These advancements will likely provide even more support for scientific research and foster greater innovation.

The journey of integrating and developing platforms like PROFASA and ChemFlow represents a significant leap forward in the field of bioinformatics. As these platforms evolve, they could play a pivotal role in shaping the future of scientific research, enabling more rapid and profound discoveries in protein design and beyond. This development reflects a broader trend towards more integrated, user-friendly, and powerful research tools that promise to expand the horizons of what researchers can achieve.

Chapter 7

Conclusion and Future Directions

This chapter is the concluding chapter of the thesis. It summarises the key findings of each chapter and suggests areas where further technological and methodological advancements are expected to drive progress in the context of protein design and medical science.

7.1 Summary of Key Findings

The thesis has explored the integration of AI within the field of bioinformatics, with a focus on enhancing the drug discovery process. The research conducted has yielded several key findings that underline the transformative potential of AI in protein design and pharmaceutical development. These findings provide a comprehensive perspective on the capabilities of machine learning technologies to address both existing and emerging challenges in bioinformatics. Here, we summarise the principal findings from the individual research themes explored.

Chapter 2 established the foundational knowledge necessary for the thesis, covering critical biological concepts like protein structure and the drug development process alongside core computational strategies, including various machine learning algorithms. This background is essential for understanding the subsequent application of these technologies in protein design.

Chapter 3 detailed how AlphaFold and AlphaFold2 have significantly advanced the field of protein structure prediction. These technologies have redefined predictive accuracy benchmarks and expanded our capabilities to understand and manipulate protein structures effectively, with AlphaFold's CNN approach and AlphaFold2's dual-lane transformer marking substantial advancements.

Chapter 4 explored the parallels between solving jigsaw puzzles and molecular recognition, offering a novel perspective on identifying and aligning interaction parts in protein design. This analogy further highlighted how deep learning can enhance the precision of drug design and improve virtual screening through the accurate prediction of protein-to-protein interaction and molecular recognition.

Chapter 5 focused on the roles of SilverSurfer and PS-GO in revolutionising parametric protein design. SilverSurfer's genetic algorithms navigate the protein space to generate novel proteins with specific functionalities, while PS-GO optimises the

search process in extensive protein databases, demonstrating how these tools collectively enhance the efficiency of parametric protein design.

Chapter 6 described the contributions of PROFASA and ChemFlow, platforms that transform protein design by automating complex computational processes, making advanced techniques accessible to a broader scientific community. This chapter emphasised the importance of user-friendly interfaces and workflow automation in democratising access to cutting-edge research tools.

7.2 Future Direction

As the field of bioinformatics and AI-assisted drug discovery continues to evolve, there are several promising avenues for future research and development, as identified throughout this thesis. The potential for significant advancements in protein structure prediction, molecular recognition, parametric protein design, and computational platforms offers a roadmap for the next generation of innovations in protein design and pharmaceutical development.

7.2.1 Protein Strcture Prediction

Future research in protein structure prediction will likely continue to leverage the properties of Deep Neural Networks, as seen in large language models, where performance seems to increase linearly with model size and data volume. Emulating the approach of LLMs, foundational models for protein folding could be developed that are capable of generalising across a variety of biophysical tasks, including the challenging prediction of membrane proteins. The integration of pertaining strategies using molecular dynamics simulations may address the current limitations posed by sparse protein structure data. Efforts to enhance AI explainability in these models will be crucial for improving their reliability and applicability in drug development and other biotechnological fields.

7.2.2 Molecular Recognition

The impact of AlphaFold on understanding protein structures has already been transformative for drug design. Future directions will likely focus on further automating drug discovery workflows using AlphaFold's structural predictions and embeddings to score interactions between drug molecules and target proteins. MRD will continue to evolve, utilising generative models to design new ligands or modify proteins for enhanced efficacy, shortening the drug discovery process and improving targeting precision. Advanced modelling techniques will aim to transform drug design by accelerating discovery processes and enhancing therapeutic interventions.

7.2.3 Parametric Protein Design

In parametric protein design, the ongoing development of tools like SilverSurfer and PS-GO will focus on optimising algorithms for navigating protein space more efficiently. Future enhancements will likely involve refining genetic algorithms and the integration of more sophisticated machine learning techniques to improve the specificity and functionality of designed proteins, potentially through the use of continuous embedding spaces generated by auto-encoder architectures. This area holds the promise of significantly advancing our ability to create novel proteins with desired properties, potentially leading to breakthroughs in various applications from medicine to industrial biotechnology.

7.2.4 Protein Platform

The future of protein platforms such as ChemFlow and PROFASA will involve not only enhancing the user experience but also expanding platform capabilities. The integration of additional computational tools and databases will enhance versatility and address specialised areas. An integrated library of generic workflows in ChemFlow, coupled with the implementation of eager execution strategies, will increase data processing efficiency and provide users with standardised, optimised options for common tasks in protein design. The potential integration of LLMs to automate Docker instance deployment and configuration could offer unprecedented flexibility and richness of integrated tools, keeping these platforms at the forefront of technological advancements in protein design.

7.2.5 General Outlook

The intersection of AI and bioinformatics as applied to protein design is poised for exponential growth. By addressing these future directions, researchers can continue to push the boundaries of what is possible in drug discovery and protein engineering, paving the way for more rapid and effective solutions to some of the most pressing health challenges facing humanity today.

Bibliography

[1] Tunde Aderinwale et al. “Real-time structure search and structure classification for AlphaFold protein models”. In: *Communications Biology* 5.1 (2022). Cited by: 26; All Open Access, Gold Open Access, Green Open Access. DOI: 10.1038/s42003-022-03261-8. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85127657577&doi=10.1038%2fs42003-022-03261-8&partnerID=40&md5=997cc9022c4281830afe453fec73fbf8.

[2] Ruedi Aebersold and Matthias Mann. “Mass-spectrometric exploration of proteome structure and function”. In: *Nature* 537.7620 (2016), pp. 347–355. ISSN: 1476-4687. DOI: 10.1038/nature19949. URL: https://doi.org/10.1038/nature19949.

[3] Abien Fred Agarap. “Deep Learning using Rectified Linear Units (ReLU)”. In: *ArXiv* abs/1803.08375 (2018). URL: https://api.semanticscholar.org/CorpusID:4090379.

[4] Shams Ahmed et al. “Deep learning modelling techniques: current progress, applications, advantages, and challenges”. In: *Artificial Intelligence Review* 56 (Apr. 2023). DOI: 10.1007/s10462-023-10466-8.

[5] M. Aizerman, E. Braverman, and L. Rozonoer. “Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning”. In: *Automation and Remote Control* 25 (1964), pp. 821–837.

[6] Aisha Al-Janabi. “Has DeepMind’s AlphaFold Solved the Protein Folding Problem?” In: *BioTechniques* 72.3 (2022). PMID: 35118884, pp. 73–76. DOI: 10.2144/btn-2022-0007. eprint: https://doi.org/10.2144/btn-2022-0007. URL: https://doi.org/10.2144/btn-2022-0007.

[7] Guillaume Alain and Yoshua Bengio. “What regularized auto-encoders learn from the data-generating distribution”. In: *J. Mach. Learn. Res.* 15.1 (Jan. 2014), 3563–3593. ISSN: 1532-4435.

[8] Rebecca F. Alford et al. “The Rosetta All-Atom Energy Function for Macromolecular Modeling and Design”. In: *J. Chem. Theory Comput.* 13.6 (June 2017), pp. 3031–3048. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.7b00125. URL: https://doi.org/10.1021/acs.jctc.7b00125.

[9] Mohammed AlQuraishi. “AlphaFold at CASP13”. In: *Bioinformatics* 35 (2019), pp. 4862–4865. DOI: 10.1093/bioinformatics/btz422.

[10] Mohammed AlQuraishi. "End-to-End Differentiable Learning of Protein Structure". In: *Cell Systems* 8.4 (2019), 292–301.e3. DOI: 10.1016/j.cels.2019.03.006.

[11] S. F. Altschul et al. "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." eng. In: *Nucleic acids research* 25 (17 Sept. 1997), pp. 3389–402. DOI: 10.1093/nar/25.17.3389.

[12] Laith Alzubaidi et al. "Novel Transfer Learning Approach for Medical Imaging with Limited Labeled Data". In: *Cancers* 13.7 (2021). ISSN: 2072-6694. DOI: 10.3390/cancers13071590. URL: https://www.mdpi.com/2072-6694/13/7/1590.

[13] Rommie E. Amaro et al. "Ensemble Docking in Drug Discovery." eng. In: *Biophysical journal* 114 (10 May 2018), pp. 2271–2278. DOI: 10.1016/j.bpj.2018.02.038.

[14] Shalini Andersson et al. "Making medicinal chemistry more effective—application of Lean Sigma to improve processes, speed and quality". In: *Drug Discovery Today* 14.11 (2009), pp. 598–604. ISSN: 1359-6446. DOI: https://doi.org/10.1016/j.drudis.2009.03.005. URL: https://www.sciencedirect.com/science/article/pii/S1359644609000750.

[15] Yi-Ru Bai et al. "A comprehensive review of small molecule drugs approved by the FDA in 2023: Advances and prospects". In: *European Journal of Medicinal Chemistry* 276 (2024), p. 116706. ISSN: 0223-5234. DOI: 10.1016/j.ejmech.2024.116706. URL: https://www.sciencedirect.com/science/article/pii/S0223523424005865.

[16] Dana H. Ballard. "Modular learning in neural networks". In: *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 1*. AAAI'87. Seattle, Washington: AAAI Press, 1987, 279–284. ISBN: 0934613427.

[17] Protein Data Bank. 2021. URL: https://www.rcsb.org/3d-view/6WVX.

[18] Atılım Günes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. "Automatic differentiation in machine learning: a survey". In: *J. Mach. Learn. Res.* 18.1 (Jan. 2017), 5595–5637. ISSN: 1532-4435.

[19] Iz Beltagy, Matthew E. Peters, and Arman Cohan. *Longformer: The Long-Document Transformer*. 2020. arXiv: 2004.05150 [cs.CL].

[20] Y. Bengio, P. Simard, and P. Frasconi. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166. DOI: 10.1109/72.279181.

[21] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. "Generalized denoising auto-encoders as generative models". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'13. Red Hook, NY, USA: Curran Associates Inc., 2013, 899–907.

[22] Helen M. Berman et al. "The Protein Data Bank". In: *Nucleic Acids Research* 28.1 (Jan. 2000), pp. 235–242. ISSN: 0305-1048. DOI: 10.1093/nar/28.1.235. eprint: https://academic.oup.com/nar/article-pdf/28/1/235/9895144/280235.pdf. URL: https://doi.org/10.1093/nar/28.1.235.

[23] Quentin Bertrand, Avishek Joey Bose, Alexandre Duplessis, Marco Jiralerspong, and Gauthier Gidel. "On the Stability of Iterative Retraining of Generative Models on their own Data". In: (2024). arXiv: 2310.00429 [cs.LG].

[24] Aman Bhargava, Cameron Witkowski, Manav Shah, and Matt Thomson. "What's the Magic Word? A Control Theory of LLM Prompting". In: *arXiv preprint arXiv:2310.04444* (2023). DOI: https://doi.org/10.48550/arXiv.2310.04444.

[25] Gavin Bierman, Martín Abadi, and Mads Torgersen. "Understanding TypeScript". In: *ECOOP 2014 – Object-Oriented Programming*. Ed. by Richard Jones. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 257–281. ISBN: 978-3-662-44202-9.

[26] Tom L. Blundell, Harren Jhoti, and Chris Abell. "High-throughput crystallography for lead discovery in drug design". In: *Nature Reviews Drug Discovery* 1.1 (2002), pp. 45–54. ISSN: 1474-1784. DOI: 10.1038/nrd706. URL: https://doi.org/10.1038/nrd706.

[27] Joel R. Bock and David A. Gough. "Predicting protein–protein interactions from primary structure". In: *Bioinformatics* 17.5 (May 2001), pp. 455–460. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/17.5.455. eprint: https://academic.oup.com/bioinformatics/article-pdf/17/5/455/48836919/bioinformatics_17_5_455.pdf. URL: https://doi.org/10.1093/bioinformatics/17.5.455.

[28] Matyas Bohacek and Hany Farid. "Nepotistically Trained Generative-AI Models Collapse". In: (2023). arXiv: 2311.12202 [cs.AI].

[29] Elliot Bolton et al. "BioMedLM: A 2.7B Parameter Language Model Trained On Biomedical Text". In: (2024). DOI: https://doi.org/10.48550/arXiv.2403.18421. arXiv: 2403.18421 [cs.CL].

[30] Aislyn D. W. Boran and Ravi Iyengar. "Systems approaches to polypharmacology and drug discovery." eng. In: *Current opinion in drug discovery & development* 13 (3 May 2010), pp. 297–309.

[31] H. Bourlard and Y. Kamp. "Auto-association by multilayer perceptrons and singular value decomposition". In: *Biological Cybernetics* 59.4 (1988), pp. 291–294. ISSN: 1432-0770. DOI: 10.1007/BF00332918. URL: https://doi.org/10.1007/BF00332918.

[32] Fergus Boyles, Charlotte M Deane, and Garrett M Morris. "Learning from the ligand: using ligand-based features to improve binding affinity prediction". In: *Bioinformatics* 36.3 (Aug. 2019), pp. 758–764. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz665. eprint: https://academic.oup.com/bioinformatics/article-pdf/36/3/758/48981883/bioinformatics_36_3_758.pdf. URL: https://doi.org/10.1093/bioinformatics/btz665.

[33] W.L. Bragg. "2 - The Diffraction of Short Electromagnetic Waves by a Crystal**Proceedings of the Cambridge Philosophical Society, 17, 43–57 (1913). Communicated by Professor Sir J. J. Thomson. Read 11 November 1912." In: *X-ray and Neutron Diffraction*. Ed. by G.E. BACON. The Commonwealth and International Library: Selected Readings in Physics. Pergamon, 1966, pp. 109–125. ISBN: 978-0-08-011999-1. DOI: https://doi.org/10.1016/B978-0-08-011999-1.50015-8. URL: https://www.sciencedirect.com/science/article/pii/B9780080119991500158.

[34] Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. "ProteinBERT: a universal deep-learning model of protein sequence and function". In: *Bioinformatics* 38.8 (Feb. 2022), pp. 2102–2110. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btac020. eprint: https://academic.oup.com/bioinformatics/article-pdf/38/8/2102/49009610/btac020.pdf. URL: https://doi.org/10.1093/bioinformatics/btac020.

[35] Leo Breiman. "Random Forests". In: *Mach. Learn.* 45.1 (Oct. 2001), 5–32. ISSN: 0885-6125. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

[36] Sergey Brin and Lawrence Page. "The anatomy of a large-scale hypertextual Web search engine". In: *Computer Networks and ISDN Systems* 30.1 (1998). Proceedings of the Seventh International World Wide Web Conference, pp. 107–117. ISSN: 0169-7552. DOI: https://doi.org/10.1016/S0169-7552(98)00110-X. URL: https://www.sciencedirect.com/science/article/pii/S016975529800110X.

[37] Luciano Brocchieri and Samuel Karlin. "Protein length in eukaryotic and prokaryotic proteomes". In: *Nucleic Acids Res.* 33.10 (2005), pp. 3390–3400.

[38] David Brown. "Unfinished business: target-based drug discovery". In: *Drug Discovery Today* 12.23 (2007), pp. 1007–1012. ISSN: 1359-6446. DOI: https://doi.org/10.1016/j.drudis.2007.10.017. URL: https://www.sciencedirect.com/science/article/pii/S1359644607004576.

[39] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].

[40] Sébastien Bubeck et al. "Sparks of Artificial General Intelligence: Early experiments with GPT-4". In: (Mar. 2023). DOI: https://doi.org/10.48550/arXiv.2303.12712. URL: https://www.microsoft.com/en-us/research/

publication/sparks-of-artificial-general-intelligence-early-experiments-with-gpt-4/.

[41] Yue Cao and Yang Shen. "TALE: Transformer-based protein function Annotation with joint sequence–Label Embedding". In: *Bioinformatics* 37.18 (Mar. 2021), pp. 2825–2833. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab198. eprint: https://academic.oup.com/bioinformatics/article-pdf/37/18/2825/50579661/btab198.pdf. URL: https://doi.org/10.1093/bioinformatics/btab198.

[42] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-End Object Detection with Transformers". In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm. Cham: Springer International Publishing, 2020, pp. 213–229. ISBN: 978-3-030-58452-8. DOI: https://doi.org/10.1007/978-3-030-58452-8_13.

[43] Paul J. Carter and Greg A. Lazar. "Next generation antibody drugs: pursuit of the 'high-hanging fruit'". In: *Nature Reviews Drug Discovery* 17.3 (Dec. 2017), pp. 197–223. DOI: 10.1038/nrd.2017.227. URL: https://doi.org/10.1038/nrd.2017.227.

[44] CASP. *Home - Prediction Center — predictioncenter.org*. https://predictioncenter.org. [Accessed 20-08-2024]. 2024.

[45] Lidia Ceriani and Paolo Verme. "The origins of the Gini index: extracts from Variabilità e Mutabilità (1912) by Corrado Gini". In: *The Journal of Economic Inequality* 10.3 (June 2011), 421–443. ISSN: 1573-8701. DOI: 10.1007/s10888-011-9188-x. URL: http://dx.doi.org/10.1007/s10888-011-9188-x.

[46] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. "A survey on adversarial attacks and defences". In: *CAAI Transactions on Intelligence Technology* 6.1 (2021), pp. 25–45. DOI: https://doi.org/10.1049/cit2.12028. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/cit2.12028. URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cit2.12028.

[47] Jean-Pierre Changeux and Stuart Edelstein. "Conformational selection or induced fit? 50 years of debate resolved." eng. In: *F1000 biology reports* 3 (2011), p. 19. DOI: 10.3410/B3-19.

[48] Jiarui Chen and Shirely W. I. Siu. "Machine Learning Approaches ofr Quality Assessment of Protein Structures". In: *Biomolecules* 10 (2020). DOI: 10.3390/biom10040626.

[49] Jiaxiao Chen, Zhonghui Gu, Luhua Lai, and Jianfeng Pei. "In silico protein function prediction: the rise of machine learning-based approaches." eng. In: *Medical review (2021)* 3 (6 2023), pp. 487–510. DOI: 10.1515/mr-2023-0038.

[50] Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey E. Hinton. "Pix2seq: A Language Modeling Framework for Object Detection". In: *ArXiv* abs/2109.10852 (2021). DOI: https://doi.org/10.48550/arXiv.2109.10852. URL: https://api.semanticscholar.org/CorpusID:237593074.

[51] Andrei Chertkov and Ivan Oseledets. "Tensor Train Decomposition for Adversarial Attacks on Computer Vision Models". In: (2023). arXiv: 2312.12556 [math.NA].

[52] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. "Generating Long Sequences with Sparse Transformers". In: *ArXiv* abs/1904.10509 (2019). URL: https://api.semanticscholar.org/CorpusID:129945531.

[53] Yu-Chiao Chiu et al. "Predicting drug response of tumors from integrated genomic profiles by deep neural networks". In: *BMC Medical Genomics* 12.1 (2019), p. 18. ISSN: 1755-8794. DOI: 10.1186/s12920-018-0460-9. URL: https://doi.org/10.1186/s12920-018-0460-9.

[54] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Conference on Empirical Methods in Natural Language Processing*. 2014. URL: https://api.semanticscholar.org/CorpusID:5590763.

[55] Taeg Sang Cho, Shai Avidan, and William T. Freeman. "A probabilistic image jigsaw puzzle solver". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 183–190. DOI: 10.1109/CVPR.2010.5540212.

[56] Krzysztof Choromanski et al. "Rethinking Attention with Performers". In: *ArXiv* abs/2009.14794 (2020). URL: https://api.semanticscholar.org/CorpusID:222067132.

[57] Aakanksha Chowdhery et al. "PaLM: Scaling Language Modeling with Pathways". In: (2022). arXiv: 2204.02311 [cs.CL].

[58] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. DOI: 10.48550/ARXIV.1412.3555. URL: https://arxiv.org/abs/1412.3555.

[59] Nicholas A. Church and John L. McKillip. "Antibiotic resistance crisis: challenges and imperatives". In: *Biologia* 76.5 (2021), pp. 1535–1550. ISSN: 1336-9563. DOI: 10.1007/s11756-021-00697-x. URL: https://doi.org/10.1007/s11756-021-00697-x.

[60] The UniProt Consortium. "UniProt: the Universal Protein Knowledgebase in 2023". In: *Nucleic Acids Research* 51.D1 (Nov. 2022), pp. D523–D531. ISSN: 0305-1048. DOI: 10.1093/nar/gkac1052. eprint: https://academic.oup.com/nar/article-pdf/51/D1/D523/48441158/gkac1052.pdf. URL: https://doi.org/10.1093/nar/gkac1052.

[61] Elias James Corey. "The Logic of Chemical Synthesis: Multistep Synthesis of Complex Carbogenic Molecules (Nobel Lecture)". In: *Angewandte Chemie International Edition in English* 30.5 (1991), pp. 455–465. DOI: https://doi.org/10.1002/anie.199104553. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/anie.199104553. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.199104553.

[62] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine Learning* 20.3 (Sept. 1995), 273–297. ISSN: 1573-0565. DOI: 10.1007/bf00994018. URL: http://dx.doi.org/10.1007/BF00994018.

[63] Thomas E. CREIGHTON. "Protein Folding". In: *Biochemistry Journal* 270 (1990), pp. 1–16.

[64] Michael P Cummings and Mark R Segal. "Few amino acid positions in rpoB are associated with most of the rifampin resistance in Mycobacterium tuberculosis". In: *BMC Bioinformatics* 5.1 (Sept. 2004). ISSN: 1471-2105. DOI: 10.1186/1471-2105-5-137. URL: http://dx.doi.org/10.1186/1471-2105-5-137.

[65] Gabriela Czibula, Alexandra-Ioana Albu, Maria Iuliana Bocicor, and Camelia Chira. "AutoPPI: An Ensemble of Deep Autoencoders for Protein–Protein Interaction Prediction". In: *Entropy* 23.6 (2021). ISSN: 1099-4300. DOI: 10.3390/e23060643. URL: https://www.mdpi.com/1099-4300/23/6/643.

[66] Sø ren Dahlgaard, Mathias Knudsen, and Mikkel Thorup. "Practical Hash Functions for Similarity Estimation and Dimensionality Reduction". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/62dad6e273d32235ae02b7d321578ee8-Paper.pdf.

[67] Brian Danchilla. "Three.js Framework". In: *Beginning WebGL for HTML5*. Berkeley, CA: Apress, 2012, pp. 173–203. ISBN: 978-1-4302-3997-0. DOI: 10.1007/978-1-4302-3997-0_7. URL: https://doi.org/10.1007/978-1-4302-3997-0_7.

[68] Elena K. Davydova. "Protein Engineering: Advances in Phage Display for Basic Science and Medical Research". In: *Biochemistry (Moscow)* 87.S1 (Jan. 2022), S146–S167. DOI: 10.1134/S0006297922140127. URL: https://doi.org/10.1134/s0006297922140127.

[69] Erik D Demaine and Martin L Demaine. "Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity". en. In: *Graphs Comb.* 23.S1 (June 2007), pp. 195–208. DOI: 10.1007/s00373-007-0713-4.

[70] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *North American Chapter of the Association for Computational Linguistics*. 2019. URL: https://api.semanticscholar.org/CorpusID:52967399.

[71] Richa Dhanuka, Anushree Tripathi, and Jyoti P. Singh. "A Semi-Supervised Autoencoder-Based Approach for Protein Function Prediction". In: *IEEE Journal of Biomedical and Health Informatics* 26.10 (2022), pp. 4957–4965. DOI: 10.1109/JBHI.2022.3163150.

[72] Joseph A. DiMasi, Henry G. Grabowski, and Ronald W. Hansen. "Innovation in the pharmaceutical industry: New estimates of R&D costs". In: *Journal of Health Economics* 47 (2016), pp. 20–33. ISSN: 0167-6296. DOI: https://doi.org/10.1016/j.jhealeco.2016.01.012. URL: https://www.sciencedirect.com/science/article/pii/S0167629616000291.

[73] CH Ding and I Dubchak. "Multi-class protein fold recognition using support vector machines and neural networks". In: *Bioinformatics (Oxford, England)* 17.4 (Apr. 2001), 349—358. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/17.4.349. URL: https://academic.oup.com/bioinformatics/article-pdf/17/4/349/937164/170349.pdf.

[74] Wenze Ding, Kenta Nakai, and Haipeng Gong. "Protein design via deep learning". In: *Briefings in Bioinformatics* 23.3 (Mar. 2022). bbac102. ISSN: 1477-4054. DOI: 10.1093/bib/bbac102. eprint: https://academic.oup.com/bib/article-pdf/23/3/bbac102/43745475/bbac102.pdf. URL: https://doi.org/10.1093/bib/bbac102.

[75] Yiran Ding et al. "LongRoPE: Extending LLM Context Window Beyond 2 Million Tokens". In: 2024. URL: https://api.semanticscholar.org/CorpusID:267770308.

[76] Samuel Dodge and Lina Karam. "A Study and Comparison of Human and Deep Learning Recognition Performance under Visual Distortions". In: *2017 26th International Conference on Computer Communication and Networks (ICCCN)*. 2017, pp. 1–7. DOI: 10.1109/ICCCN.2017.8038465.

[77] Martin J Doherty, Marina C Wimmer, Cornelia Gollek, Charlotte Stone, and Elizabeth J Robinson. "Piecing together the puzzle of pictorial representation: How jigsaw puzzles index metacognitive development". en. In: *Child Dev.* 92.1 (Jan. 2021), pp. 205–221.

[78] Bin Dong, Fangao Zeng, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. "SOLQ: Segmenting Objects by Learning Queries". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 21898–21909. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/b7087c1f4f89e63af8d46f3b20271153-Paper.pdf.

[79] Marco Dorigo and Thomas Stützle. "Ant Colony Optimization: Overview and Recent Advances". In: *Handbook of Metaheuristics*. Ed. by Michel Gendreau and Jean-Yves Potvin. Cham: Springer International Publishing, 2019, pp. 311–351. DOI: 10.1007/978-3-319-91086-4_10. URL: https://doi.org/10.1007/978-3-319-91086-4_10.

[80] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *ArXiv* abs/2010.11929 (2020). DOI: https://doi.org/10.48550/arXiv.2010.11929. URL: https://api.semanticscholar.org/CorpusID:225039882.

[81] Bo Du, Ziyi Liu, and Fulin Luo. "Deep multi-scale attention network for RNA-binding proteins prediction". In: *Information Sciences* 582 (2022), pp. 287–301. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2021.09.025. URL: https://www.sciencedirect.com/science/article/pii/S0020025521009567.

[82] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". In: *J. Mach. Learn. Res.* 12.null (June 2011), 2121–2159. ISSN: 1532-4435.

[83] A. E. Eiben, P. E. Raué, and Zs. Ruttkay. "Genetic algorithms with multi-parent recombination". In: *Parallel Problem Solving from Nature — PPSN III*. Ed. by Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 78–87. ISBN: 978-3-540-49001-2.

[84] Ahmed Elnaggar et al. "ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (2022), pp. 7112–7127. DOI: 10.1109/TPAMI.2021.3095381.

[85] Gábor Erdős, Mátyás Pajkos, and Zsuzsanna Dosztányi. "IUPred3: prediction of protein disorder enhanced with unambiguous experimental annotation and visualization of evolutionary conservation". In: *Nucleic Acids Research* 49.W1 (May 2021), W297–W303. DOI: 10.1093/nar/gkab408. URL: https://doi.org/10.1093/nar/gkab408.

[86] Daniel A. Erlanson, Stephen W. Fesik, Roderick E. Hubbard, Wolfgang Jahnke, and Harren Jhoti. "Twenty years on: the impact of fragments on drug discovery". In: *Nature Reviews Drug Discovery* 15.9 (2016), pp. 605–619. ISSN: 1474-1784. DOI: 10.1038/nrd.2016.109. URL: https://doi.org/10.1038/nrd.2016.109.

[87] Fabio Fabris, Aoife Doherty, Daniel Palmer, João Pedro de Magalhães, and Alex A Freitas. "A new approach for interpreting Random Forest models and its application to the biology of ageing". In: *Bioinformatics* 34.14 (Feb. 2018). Ed. by Jonathan Wren, 2449–2456. ISSN: 1367-4811. DOI: 10.1093/

bioinformatics/bty087. URL: http://dx.doi.org/10.1093/bioinformatics/bty087.

[88] Zhihao Fan et al. "Mask Attention Networks: Rethinking and Strengthen Transformer". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Kristina Toutanova et al. Online: Association for Computational Linguistics, June 2021, pp. 1692–1701. DOI: 10.18653/v1/2021.naacl-main.135. URL: https://aclanthology.org/2021.naacl-main.135.

[89] FDA. *Novel Drug Approvals at FDA — fda.gov*. https://www.fda.gov/drugs/development-approval-process-drugs/novel-drug-approvals-fda. [Accessed 03-05-2024]. 2024.

[90] Yinchao Fei, Hao Zhang, Yili Wang, Zhen Liu, and Yuanning Liu. "LTPConstraint: a transfer learning based end-to-end method for RNA secondary structure prediction". In: *BMC Bioinformatics* 23.1 (2022), p. 354. ISSN: 1471-2105. DOI: 10.1186/s12859-022-04847-z. URL: https://doi.org/10.1186/s12859-022-04847-z.

[91] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. "ProtGPT2 is a deep unsupervised language model for protein design". In: *Nature Communications* 13.1 (2022), p. 4348. ISSN: 2041-1723. DOI: 10.1038/s41467-022-32007-7. URL: https://doi.org/10.1038/s41467-022-32007-7.

[92] Robert D. Finn, Jody Clements, and Sean R. Eddy. "HMMER web server: interactive sequence similarity searching". In: *Nucleic Acids Research* 39.suppl_2 (May 2011), W29–W37. ISSN: 0305-1048. DOI: 10.1093/nar/gkr367. eprint: https://academic.oup.com/nar/article-pdf/39/suppl_2/W29/7628106/gkr367.pdf. URL: https://doi.org/10.1093/nar/gkr367.

[93] Robert D. Finn et al. "The Pfam protein families database: towards a more sustainable future". In: *Nucleic Acids Research* 44.D1 (Dec. 2015), pp. D279–D285. ISSN: 0305-1048. DOI: 10.1093/nar/gkv1344. eprint: https://academic.oup.com/nar/article-pdf/44/D1/D279/9484040/gkv1344.pdf. URL: https://doi.org/10.1093/nar/gkv1344.

[94] Emil Fischer. "Einfluss der Configuration auf die Wirkung der Enzyme". In: *European Journal of Inorganic Chemistry* 27 (1894), pp. 2985–2993. URL: https://api.semanticscholar.org/CorpusID:84051139.

[95] Thomas G. Flower and James H. Hurley. "Crystallographic molecular replacement using an in silico-generated search model of SARS-CoV-2 ORF8". In: *Protein Science* 30.4 (Mar. 2021), 728–734. ISSN: 1469-896X. DOI: 10.1002/pro.4050. URL: http://dx.doi.org/10.1002/pro.4050.

[96] R. M. Friedberg. "A Learning Machine: Part I". In: *IBM Journal of Research and Development* 2.1 (1958), pp. 2–13. DOI: 10.1147/rd.21.0002.

[97] W. Friedrich, P. Knipping, and M. Laue. "Interferenzerscheinungen bei Röntgenstrahlen". In: *Annalen der Physik* 346.10 (1913), pp. 971–988. DOI: https://doi.org/10.1002/andp.19133461004. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/andp.19133461004. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/andp.19133461004.

[98] Stephan Fuchs. *Gravy calculator*. [Online; accessed 14-March-2023]. URL: http://gravy-calculator.de/.

[99] Kunihiko Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics* 36.4 (Apr. 1980), 193–202. ISSN: 1432-0770. DOI: 10.1007/bf00344251. URL: http://dx.doi.org/10.1007/BF00344251.

[100] Andrew C. Gallagher. "Jigsaw puzzles with pieces of unknown orientation". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 382–389. DOI: 10.1109/CVPR.2012.6247699.

[101] Elisabeth Gasteiger et al. "Protein Identification and Analysis Tools on the ExPASy Server". In: *The Proteomics Protocols Handbook*. Ed. by John M. Walker. Totowa, NJ: Humana Press, 2005, pp. 571–607. ISBN: 978-1-59259-890-8. DOI: 10.1385/1-59259-890-0:571. URL: https://doi.org/10.1385/1-59259-890-0:571.

[102] Robert Gaynes. "The Discovery of Penicillin–New Insights After More Than 75 Years of Clinical Use." eng. In: *Emerging Infectious Diseases* 23 (5 May 2017), pp. 849–53. DOI: 10.3201/eid2305.161556.

[103] Pierre Geurts et al. "Proteomic mass spectra classification using decision tree based ensemble methods". In: *Bioinformatics* 21.14 (May 2005), pp. 3138–3145. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/bti494. eprint: https://academic.oup.com/bioinformatics/article-pdf/21/14/3138/48971458/bioinformatics_21_14_3138.pdf. URL: https://doi.org/10.1093/bioinformatics/bti494.

[104] Stefano Gianni, Jakob Dogan, and Per Jemth. "Distinguishing induced fit from conformational selection." eng. In: *Biophysical chemistry* 189 (May 2014), pp. 33–9. DOI: 10.1016/j.bpc.2014.03.003.

[105] Vladimir Gligorijević et al. "Structure-based protein function prediction using graph convolutional networks". In: *Nature Communications* 12.1 (2021). Cited by: 271; All Open Access, Gold Open Access. DOI: 10.1038/s41467-021-23303-9. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85106878177&doi=10.1038%2fs41467-021-23303-9&partnerID=40&md5=dee3daa4ddd53c00ac4985ed22153a38.

[106] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[107] Jiaqi Gu et al. "Multi-Scale High-Resolution Vision Transformer for Semantic Segmentation". In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 12084–12093. DOI: 10.1109/CVPR52688.2022.01178.

[108] Abhishek Gupta, Alagan Anpalagan, Ling Guan, and Ahmed Shaharyar Khwaja. "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues". In: *Array* 10 (2021), p. 100057. ISSN: 2590-0056. DOI: https://doi.org/10.1016/j.array.2021.100057. URL: https://www.sciencedirect.com/science/article/pii/S2590005621000059.

[109] Namita T. Gupta, Jason A. Vander Heiden, Mohamed Uduman, Daniel Gadala-Maria, Gur Yaari, and Steven H. Kleinstein. "Change-O: a toolkit for analyzing large-scale B cell immunoglobulin repertoire sequencing data". In: *Bioinformatics* 31.20 (June 2015), pp. 3356–3358. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btv359. eprint: https://academic.oup.com/bioinformatics/article-pdf/31/20/3356/49035468/bioinformatics_31_20_3356.pdf. URL: https://doi.org/10.1093/bioinformatics/btv359.

[110] Shir Gur and Ohad Ben-Shahar. "From Square Pieces to Brick Walls: The Next Challenge in Solving Jigsaw Puzzles". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 4049–4057. DOI: 10.1109/ICCV.2017.434.

[111] Kunchur Guruprasad, B.V.Bhasker Reddy, and Madhusudan W. Pandit. "Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence". In: *Protein Engineering, Design and Selection* 4.2 (Dec. 1990), pp. 155–161. ISSN: 1741-0126. DOI: 10.1093/protein/4.2.155. eprint: https://academic.oup.com/peds/article-pdf/4/2/155/4316164/4-2-155.pdf. URL: https://doi.org/10.1093/protein/4.2.155.

[112] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. "Gene Selection for Cancer Classificationusing Support Vector Machines". In: *Machine Learning* 46.1/3 (2002), 389–422. ISSN: 0885-6125. DOI: 10.1023/a:1012487302797. URL: http://dx.doi.org/10.1023/A:1012487302797.

[113] Rafael Gómez-Bombarelli et al. "Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules". In: *ACS Central Science* 4.2 (2018). PMID: 29532027, pp. 268–276. DOI: 10.1021/acscentsci.7b00572. eprint: https://doi.org/10.1021/acscentsci.7b00572. URL: https://doi.org/10.1021/acscentsci.7b00572.

[114] Philip J. Hajduk and Jonathan Greer. "A decade of fragment-based drug design: strategic advances and lessons learned". In: *Nature Reviews Drug Discovery* 6.3 (2007), pp. 211–219. ISSN: 1474-1784. DOI: 10.1038/nrd2220. URL: https://doi.org/10.1038/nrd2220.

[115] William L. Hamilton, Rex Ying, and Jure Leskovec. "Inductive representation learning on large graphs". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, 1025–1035. ISBN: 9781510860964.

[116] Boris Hanin and David Rolnick. "How to start training: the effect of initialization and architecture". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, 569–579.

[117] Richard K. Harrison. "Phase II and phase III failures: 2013-2015". In: *Nature Reviews Drug Discovery* 15.12 (2016), pp. 817–818. ISSN: 1474-1784. DOI: 10.1038/nrd.2016.184. URL: https://doi.org/10.1038/nrd.2016.184.

[118] Sue Ellen Haupt et al. "The History and Practice of AI in the Environmental Sciences". In: *Bulletin of the American Meteorological Society* 103.5 (2022), E1351 –E1370. DOI: 10.1175/BAMS-D-20-0234.1. URL: https://journals.ametsoc.org/view/journals/bams/103/5/BAMS-D-20-0234.1.xml.

[119] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[120] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV '15. USA: IEEE Computer Society, 2015, 1026–1034. ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.123. URL: https://doi.org/10.1109/ICCV.2015.123.

[121] Thao Thi Ho et al. "A 3D-CNN model with CT-based parametric response mapping for classifying COPD subjects". In: *Scientific Reports* 11.1 (Jan. 2021). ISSN: 2045-2322. DOI: 10.1038/s41598-020-79336-5. URL: http://dx.doi.org/10.1038/s41598-020-79336-5.

[122] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*. Vol. 1. 1995, 278–282 vol.1. DOI: 10.1109/ICDAR.1995.598994.

[123] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[124] Joshua Hochuli, Alec Helbling, Tamar Skaist, Matthew Ragoza, and David Ryan Koes. "Visualizing convolutional neural network protein-ligand scoring". In: *Journal of Molecular Graphics and Modelling* 84 (2018), pp. 96–108. ISSN: 1093-3263. DOI: https://doi.org/10.1016/j.jmgm.2018.06.005. URL: https://www.sciencedirect.com/science/article/pii/S1093326318301670.

[125] John H Holland. *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press, Dec. 1975.

[126] Liisa Holm and Laura M. Laakso. "Dali server update". In: *Nucleic Acids Research* 44.W1 (Apr. 2016), W351–W355. ISSN: 0305-1048. DOI: 10.1093/nar/gkw357. eprint: https://academic.oup.com/nar/article-pdf/44/W1/W351/7633250/gkw357.pdf. URL: https://doi.org/10.1093/nar/gkw357.

[127] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5 (1989), pp. 359–366. ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(89)90020-8. URL: https://www.sciencedirect.com/science/article/pii/0893608089900208.

[128] Yunfei Hu et al. "NMR-Based Methods for Protein Analysis". In: *Anal. Chem.* 93.4 (Feb. 2021), pp. 1866–1879. ISSN: 0003-2700. DOI: 10.1021/acs.analchem.0c03830. URL: https://doi.org/10.1021/acs.analchem.0c03830.

[129] Sujun Hua and Zhirong Sun. "A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach". In: *Journal of Molecular Biology* 308.2 (2001), pp. 397–407. ISSN: 0022-2836. DOI: https://doi.org/10.1006/jmbi.2001.4580. URL: https://www.sciencedirect.com/science/article/pii/S0022283601945802.

[130] Kexin Huang, Cao Xiao, Lucas M Glass, and Jimeng Sun. "MolTrans: Molecular Interaction Transformer for drug–target interaction prediction". In: *Bioinformatics* 37.6 (Oct. 2020), pp. 830–836. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btaa880. eprint: https://academic.oup.com/bioinformatics/article-pdf/37/6/830/50357347/btaa880_supplementary_data.pdf. URL: https://doi.org/10.1093/bioinformatics/btaa880.

[131] Yuyao Huang, Jiesi Luo, Runyu Jing, and Menglong Li. "Multi-model predictive analysis of RNA solvent accessibility based on modified residual attention mechanism". In: *Briefings in Bioinformatics* 23.6 (Oct. 2022), bbac470. ISSN: 1477-4054. DOI: 10.1093/bib/bbac470. eprint: https://academic.oup.com/bib/article-pdf/23/6/bbac470/47143923/bbac470.pdf. URL: https://doi.org/10.1093/bib/bbac470.

[132] JP Hughes, S Rees, SB Kalindjian, and KL Philpott. "Principles of early drug discovery". eng. In: *British Journal of Pharmacology* 162.6 (6 Mar. 2011), pp. 1239–1249. DOI: https://doi.org/10.1111/j.1476-5381.2010.01127.x. eprint: https://bpspubs.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1476-5381.2010.01127.x. URL: https://bpspubs.onlinelibrary.wiley.com/doi/abs/10.1111/j.1476-5381.2010.01127.x.

[133] Fergus Imrie, Anthony R. Bradley, Mihaela van der Schaar, and Charlotte M. Deane. "Protein Family-Specific Models Using Deep Neural Networks and Transfer Learning Improve Virtual Screening and Highlight the Need

for More Data". In: *Journal of Chemical Information and Modeling* 58.11 (Oct. 2018), 2319–2330. ISSN: 1549-960X. DOI: 10.1021/acs.jcim.8b00350. URL: http://dx.doi.org/10.1021/acs.jcim.8b00350.

[134] Saidul Islam et al. "A comprehensive survey on applications of transformers for deep learning tasks". In: *Expert Systems with Applications* 241 (2024), p. 122666. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.122666. URL: https://www.sciencedirect.com/science/article/pii/S0957417423031688.

[135] Tommi Jaakkola, Mark Diekhans, and David Haussler. "A Discriminative Framework for Detecting Remote Protein Homologies". In: *Journal of Computational Biology* 7.1-2 (2000). PMID: 10890390, pp. 95–114. DOI: 10.1089/10665270050081405. eprint: https://doi.org/10.1089/10665270050081405. URL: https://doi.org/10.1089/10665270050081405.

[136] Tommi S. Jaakkola and David Haussler. "Exploiting generative models in discriminative classifiers". In: *Proceedings of the 11th International Conference on Neural Information Processing Systems*. NIPS'98. Denver, CO: MIT Press, 1998, 487–493.

[137] Sabrina Jaeger, Simone Fulle, and Samo Turk. "Mol2vec: Unsupervised Machine Learning Approach with Chemical Intuition". In: *J. Chem. Inf. Model.* 58.1 (Jan. 2018), pp. 27–35. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.7b00616. URL: https://doi.org/10.1021/acs.jcim.7b00616.

[138] Brankica G. Jankovic and Natalija D. Polovic. "The protein folding problem". In: *Biologia Serbica* 39 (2017), pp. 105–111. DOI: 10.5281/zenodo.827151.

[139] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. "DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome". In: *Bioinformatics* 37.15 (Feb. 2021), pp. 2112–2120. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab083. eprint: https://academic.oup.com/bioinformatics/article-pdf/37/15/2112/57195892/btab083.pdf. URL: https://doi.org/10.1093/bioinformatics/btab083.

[140] Ziwei Ji et al. "Survey of Hallucination in Natural Language Generation". In: *ACM Comput. Surv.* 55.12 (Mar. 2023). ISSN: 0360-0300. DOI: 10.1145/3571730. URL: https://doi.org/10.1145/3571730.

[141] José Jiménez, Miha Škalič, Gerard Martínez-Rosell, and Gianni De Fabritiis. "KDEEP: Protein-Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks". In: *J. Chem. Inf. Model.* 58.2 (Feb. 2018), pp. 287–296. ISSN: 1549-9596. DOI: 10.1021/acs.jcim.7b00650. URL: https://doi.org/10.1021/acs.jcim.7b00650.

[142] Qiao Jin, Yifan Yang, Qingyu Chen, and Zhiyong Lu. "GeneGPT: augmenting large language models with domain tools for improved access to biomedical information". In: *Bioinformatics* 40.2 (Feb. 2024), btae075. ISSN: 1367-4811. DOI: 10.1093/bioinformatics/btae075. eprint: https://academic.oup.com/bioinformatics/article-pdf/40/2/btae075/56803364/btae075.pdf. URL: https://doi.org/10.1093/bioinformatics/btae075.

[143] John Jumper et al. *AlphaFold2*. 2020. URL: https://predictioncenter.org/casp14/doc/presentations/2020_12_01_TS_predictor_AlphaFold2.pdf.

[144] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (July 2021), 583–589. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2. URL: http://dx.doi.org/10.1038/s41586-021-03819-2.

[145] Jared Kaplan et al. "Scaling Laws for Neural Language Models". In: *ArXiv* abs/2001.08361 (2020). URL: https://api.semanticscholar.org/CorpusID:210861095.

[146] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. "A review on genetic algorithm: past, present, and future." eng. In: *Multimedia tools and applications* 80 (5 2021), pp. 8091–8126. DOI: 10.1007/s11042-020-10139-6.

[147] Malgorzata Kesik-Brodacka. "Progress in biopharmaceutical development". In: *Biotechnology and Applied Biochemistry* 65.3 (Nov. 2017), pp. 306–322. DOI: 10.1002/bab.1617. URL: https://doi.org/10.1002/bab.1617.

[148] Eunyoung Kim and Hojung Nam. "DeSIDE-DDI: interpretable prediction of drug-drug interactions using drug-induced gene expressions." eng. In: *Journal of cheminformatics* 14 (1 Mar. 2022), p. 9. DOI: 10.1186/s13321-022-00589-5.

[149] Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980 (2014). URL: https://api.semanticscholar.org/CorpusID:6628106.

[150] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *CoRR* abs/1312.6114 (2013). URL: https://api.semanticscholar.org/CorpusID:216078090.

[151] Michael Schantz Klausen et al. "NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning". In: *Proteins: Structure, Function, and Bioinformatics* 87.6 (Mar. 2019), pp. 520–527. DOI: 10.1002/prot.25674. URL: https://doi.org/10.1002/prot.25674.

[152] D. E. Koshland. "Application of a Theory of Enzyme Specificity to Protein Synthesis." eng. In: *Proceedings of the National Academy of Sciences of the United States of America* 44 (2 Feb. 1958), pp. 98–104. DOI: 10.1073/pnas.44.2.98.

[153] D.A. Kosiba, P.M. Devaux, S. Balasubramanian, T.L. Gandhi, and K. Kasturi. "An automatic jigsaw puzzle solver". In: *Proceedings of 12th International Conference on Pattern Recognition*. Vol. 1. IEEE Comput. Soc. Press, 1994, 616–618 vol.1. DOI: 10.1109/ICPR.1994.576377.

[154] Lukasz P. Kozlowski. "IPC – Isoelectric Point Calculator". In: *Biology Direct* 11.1 (Oct. 2016), p. 55. ISSN: 1745-6150. DOI: 10.1186/s13062-016-0159-9. URL: https://doi.org/10.1186/s13062-016-0159-9.

[155] Jeffrey A. Kramer, John E. Sagartz, and Dale L. Morris. "The application of discovery toxicology and pathology towards the design of safer pharmaceutical lead candidates". In: *Nature Reviews Drug Discovery* 6.8 (2007), pp. 636–649. ISSN: 1474-1784. DOI: 10.1038/nrd2378. URL: https://doi.org/10.1038/nrd2378.

[156] Alex Krizhevsky and Geoffrey E. Hinton. "Using very deep autoencoders for content-based image retrieval". In: *The European Symposium on Artificial Neural Networks*. 2011. URL: https://api.semanticscholar.org/CorpusID:16907205.

[157] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Commun. ACM* 60.6 (May 2017), 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386.

[158] Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. "Critical assessment of methods of protein structure prediction (CASP)– Round XIII". In: *Proteins* 87 (2019), pp. 1011–1020. DOI: 10.1002/prot.25823.

[159] Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. "Critical assessment of methods of protein structure prediction (CASP)—Round XIV". In: *Proteins: Structure, Function, and Bioinformatics* 89.12 (2021), pp. 1607–1617. DOI: https://doi.org/10.1002/prot.26237. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.26237. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.26237.

[160] Hugo Kubinyi. "QSAR : Hansch analysis and related approaches". In: 1993. ISBN: 9783527616824. DOI: 10.1002/9783527616824. URL: https://api.semanticscholar.org/CorpusID:31487789.

[161] Brian Kuhlman and Philip Bradley. "Advances in protein structure prediction and design". In: *Nature Reviews Molecular Cell Biology* 20.11 (2019), pp. 681–697. DOI: 10.1038/s41580-019-0163-x.

[162] Maxat Kulmanov and Robert Hoehndorf. "DeepGOPlus: improved protein function prediction from sequence". In: *Bioinformatics* 36.2 (July 2019). Ed. by Lenore Cowen, pp. 422–429. DOI: 10.1093/bioinformatics/btz595. URL: https://doi.org/10.1093/bioinformatics/btz595.

[163] Roger Lahana. "How many leads from HTS?" In: *Drug Discovery Today* 4.10 (1999), pp. 447–448. ISSN: 1359-6446. DOI: https://doi.org/10.1016/S1359-6446(99)01393-8. URL: https://www.sciencedirect.com/science/article/pii/S1359644699013938.

[164] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. "Unmasking Clever Hans predictors and assessing what machines really learn". In: *Nature Communications* 10.1 (Mar. 2019). ISSN: 2041-1723. DOI: 10.1038/s41467-019-08987-4. URL: http://dx.doi.org/10.1038/s41467-019-08987-4.

[165] Anne Laure Le Page et al. "Using a convolutional neural network for classification of squamous and non-squamous non-small cell lung cancer based on diagnostic histopathology HES images". In: *Scientific Reports* 11.1 (Dec. 2021). ISSN: 2045-2322. DOI: 10.1038/s41598-021-03206-x. URL: http://dx.doi.org/10.1038/s41598-021-03206-x.

[166] Andrew R. Leach, Valerie J. Gillet, Richard A. Lewis, and Robin Taylor. "Three-Dimensional Pharmacophore Methods in Drug Discovery". In: *J. Med. Chem.* 53.2 (Jan. 2010), pp. 539–558. ISSN: 0022-2623. DOI: 10.1021/jm900817u. URL: https://doi.org/10.1021/jm900817u.

[167] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[168] Y. LeCun et al. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.

[169] Y LeCun et al. "Handwritten digit recognition with a back-propagation network, 1989". In: *Neural Information Processing Systems (NIPS)*. 1989, pp. 396–404.

[170] Ingoo Lee, Jongsoo Keum, and Hojung Nam. "DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences". In: *PLOS Computational Biology* 15.6 (June 2019). Ed. by James M. Briggs, e1007129. DOI: 10.1371/journal.pcbi.1007129. URL: https://doi.org/10.1371/journal.pcbi.1007129.

[171] Jae-Won Lee et al. "DeepFold: Enhancing protein structure prediction through optimized loss functions, improved template features, and re-optimized energy function". In: *Bioinformatics* 39.12 (2023). Cited by: 2; All Open Access, Gold Open Access. DOI: 10.1093/bioinformatics/btad712. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85179019341&doi=10.1093%2fbioinformatics%2fbtad712&partnerID=40&md5=5dd6caeba4fd76e2a7a49aad3c3ef175

[172] Sumudu P. Leelananda and Steffen Lindert. "Computational methods in drug discovery". In: *Beilstein Journal of Organic Chemistry* 12 (2016), pp. 2694–2718. ISSN: 1860-5397. DOI: 10.3762/bjoc.12.267. URL: https://doi.org/10.3762/bjoc.12.267.

[173] Julia Koehler Leman et al. "Macromolecular modeling and design in Rosetta: recent methods and frameworks". In: *Nature Methods* 17.7 (2020), pp. 665–680. ISSN: 1548-7105. DOI: 10.1038/s41592-020-0848-2. URL: https://doi.org/10.1038/s41592-020-0848-2.

[174] Christina S. Leslie, Eleazar Eskin, and William Stafford Noble. "The Spectrum Kernel: A String Kernel for SVM Protein Classification". In: *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing* (2001), pp. 564–75. URL: https://api.semanticscholar.org/CorpusID:9725578.

[175] Shawn E. Levy and Braden E. Boone. "Next-Generation Sequencing Strategies." eng. In: *Cold Spring Harbor perspectives in medicine* 9 (7 June 2019). DOI: 10.1101/cshperspect.a025791.

[176] Qingxin Li and Congbao Kang. *Perspectives on Fragment-based Drug Discovery: A Strategy Applicable to Diverse Targets*. 2021. DOI: 10.2174/1568026621666210804115700. URL: http://www.eurekaselect.com/article/117031.

[177] Zhaowen Li et al. "MST: Masked Self-Supervised Transformer for Visual Representation". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 13165–13176. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/6dbbe6abe5f14af882ff977fc3f35501-Paper.pdf.

[178] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2015. arXiv: 1405.0312 [cs.CV].

[179] Yong Lin et al. "Spurious Feature Diversification Improves Out-of-distribution Generalization". In: (2023). arXiv: 2309.17230 [cs.LG].

[180] Yun Lin, Haojun Zhao, Xuefei Ma, Ya Tu, and Meiyu Wang. "Adversarial Attacks in Modulation Recognition With Convolutional Neural Networks". In: *IEEE Transactions on Reliability* 70.1 (2021), pp. 389–401. DOI: 10.1109/TR.2020.3032744.

[181] C. A. Lipinski, F. Lombardo, B. W. Dominy, and P. J. Feeney. "Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings." eng. In: *Advanced drug delivery reviews* 46 (1-3 Mar. 2001), pp. 3–26. DOI: 10.1016/s0169-409x(00)00129-0.

[182] Zachary Chase Lipton. "A Critical Review of Recurrent Neural Networks for Sequence Learning". In: *ArXiv* abs/1506.00019 (2015). URL: https://api.semanticscholar.org/CorpusID:5837849.

[183] Chunyu Liu, H. Hoxie Ackerman, and John P. Carulli. "A genome-wide screen of gene–gene interactions for rheumatoid arthritis susceptibility". In: *Human Genetics* 129.5 (Jan. 2011), 473–485. ISSN: 1432-1203. DOI: 10.1007/s00439-010-0943-z. URL: http://dx.doi.org/10.1007/s00439-010-0943-z.

[184] Qian Liu et al. "You Impress Me: Dialogue Generation via Mutual Persona Perception". In: *Annual Meeting of the Association for Computational Linguistics*. 2020. URL: https://api.semanticscholar.org/CorpusID:215745354.

[185] Ze Liu et al. "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 10012–10022. DOI: https://doi.org/10.48550/arXiv.2103.14030.

[186] Zhaoshan Liu, Qiujie Lv, Ziduo Yang, Yifan Li, Chau Hung Lee, and Lei Shen. "Recent progress in transformer-based medical image analysis". In: *Computers in Biology and Medicine* 164 (2023), p. 107268. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2023.107268. URL: https://www.sciencedirect.com/science/article/pii/S0010482523007333.

[187] Bertram Ludäscher, Mathias Weske, Timothy McPhillips, and Shawn Bowers. "Scientific Workflows: Business as Usual?" In: *Business Process Management*. Ed. by Umeshwar Dayal, Johann Eder, Jana Koehler, and Hajo A. Reijers. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 31–47. ISBN: 978-3-642-03848-8.

[188] Renqian Luo et al. "BioGPT: generative pre-trained transformer for biomedical text generation and mining". In: *Briefings in Bioinformatics* 23.6 (Sept. 2022), bbac409. ISSN: 1477-4054. DOI: 10.1093/bib/bbac409. eprint: https://academic.oup.com/bib/article-pdf/23/6/bbac409/47144271/bbac409.pdf. URL: https://doi.org/10.1093/bib/bbac409.

[189] N. M. Luscombe, D. Greenbaum, and M. Gerstein. "What is bioinformatics? An introduction and overview". EN. In: *Yearbook of Medical Informatics* 10.01 (2001), pp. 83–100. ISSN: 2364-0502. DOI: 10.1055/s-0038-1638103. URL: http://www.thieme-connect.com/products/ejournals/abstract/10.1055/s-0038-1638103.

[190] Liuzhenghao Lv et al. "ProLLaMA: A Protein Large Language Model for Multi-Task Protein Language Processing". In: (2024). DOI: https://doi.org/10.48550/arXiv.2402.16445. arXiv: 2402.16445 [cs.CE].

[191] Ali Madani et al. "ProGen: Language Modeling for Protein Generation". In: (2020). DOI: https://doi.org/10.48550/arXiv.2004.03497. arXiv: 2004.03497 [q-bio.BM].

[192] G. Madhavi Sastry, Matvey Adzhigirey, Tyler Day, Ramakrishna Annabhimoju, and Woody Sherman. "Protein and ligand preparation: parameters, protocols, and influence on virtual screening enrichments". In: *Journal of Computer-Aided Molecular Design* 27.3 (2013), pp. 221–234. ISSN: 1573-4951. DOI: 10.1007/s10822-013-9644-8. URL: https://doi.org/10.1007/s10822-013-9644-8.

[193] Stefan-Bogdan Marcu, Yanlin Mi, Mark Tangney, and Sabin Tabirca. "ChemFlow: Bridging Computational Tools and Bioinformatics through LLM-Driven Workflow Automation Generating Jigsaw Puzzles and an AI Powered Solver". In: *Modelling and Development of Intelligent Systems*. Ed. by Dana Simian and Laura Florentina Stoica. Cham: Springer Nature Switzerland, 2024.

[194] Stefan-Bogdan Marcu, Yanlin Mi, Venkata V B Yallapragada, Mark Tangney, and Sabin Tabirca. "Silver Surfer: Navigating the Parametric Protein Space with Genetic Algorithms". In: *The 37th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems*. 2024. DOI: 10.1007/978-981-97-4677-4_39.

[195] Ştefan-Bogdan Marcu, Yanlin Mi, Venkata V. B. Yallapragada, Mark Tangney, and Sabin Tabirca. "Generating Jigsaw Puzzles and an AI Powered Solver". In: *Modelling and Development of Intelligent Systems*. Ed. by Dana Simian and Laura Florentina Stoica. Cham: Springer Nature Switzerland, 2023, pp. 148–160. ISBN: 978-3-031-27034-5.

[196] Ştefan-Bogdan Marcu, Sabin Tăbîrcă, and Mark Tangney. "An Overview of Alphafold's Breakthrough". In: *Frontiers in Artificial Intelligence* 5 (2022). ISSN: 2624-8212. DOI: 10.3389/frai.2022.875587. URL: https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2022.875587.

[197] Leandro Martínez. "Introducing the Levinthal's Protein Folding Paradox and Its Solution". In: *Chemical Education* 91 (2014), pp. 1918–1923. DOI: 10.1021/ed300302h.

[198] Andrew T. McNutt et al. "GNINA 1.0: molecular docking with deep learning". In: *Journal of Cheminformatics* 13.1 (2021), p. 43. ISSN: 1758-2946. DOI: 10.1186/s13321-021-00522-2. URL: https://doi.org/10.1186/s13321-021-00522-2.

[199] Yanlin Mi, Stefan-Bogdan Marcu, Sabin Tabirca, and Venkata V. B. Yallapragada. "PROFASA-a web-based protein fragment and structure analysis workstation." eng. In: *Frontiers in bioengineering and biotechnology* 11 (2023), p. 1192094. DOI: 10.3389/fbioe.2023.1192094.

[200] Yanlin Mi, Stefan-Bogdan Marcu, Sabin Tabirca, and Venkata V.B. Yallapragada. "PS-GO parametric protein search engine". In: *Computational and Structural Biotechnology Journal* 23 (2024), pp. 1499–1509. ISSN: 2001-0370. DOI: https:

//doi.org/10.1016/j.csbj.2024.04.003. URL: https://www.sciencedirect.com/science/article/pii/S2001037024000886.

[201] Yanlin Mi, Stefan-Bogdan Marcu, Venkata V. B. Yallapragada, and Sabin Tabirca. "ProteinFlow: An advanced framework for feature engineering in protein data analysis". In: *Biotechnology and Bioengineering* n/a.n/a (2024). DOI: https://doi.org/10.1002/bit.28812. eprint: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/bit.28812. URL: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/bit.28812.

[202] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient Estimation of Word Representations in Vector Space". In: (2013). DOI: https://doi.org/10.48550/arXiv.1301.3781. arXiv: 1301.3781 [cs.CL].

[203] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. "Image Segmentation Using Deep Learning: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.7 (2022), pp. 3523–3542. DOI: 10.1109/TPAMI.2021.3059968.

[204] Thomas Mitchell. *Machine Learning*. en. McGraw-Hill series in computer science. New York, NY: McGraw-Hill Professional, Mar. 1997.

[205] Vladimír Mlynárik. "Introduction to nuclear magnetic resonance". In: *Analytical Biochemistry* 529 (2017). Introduction to in vivo Magnetic Resonance Spectroscopy (MRS): A method to non-invasively study metabolism, pp. 4–9. ISSN: 0003-2697. DOI: https://doi.org/10.1016/j.ab.2016.05.006. URL: https://www.sciencedirect.com/science/article/pii/S0003269716300719.

[206] Marc A. Moesser, Dominik Klein, Fergus Boyles, Charlotte M. Deane, Andrew Baxter, and Garrett M. Morris. "Protein-Ligand Interaction Graphs: Learning from Ligand-Shaped 3D Interaction Graphs to Improve Binding Affinity Prediction". In: (Mar. 2022). DOI: 10.1101/2022.03.04.483012. URL: http://dx.doi.org/10.1101/2022.03.04.483012.

[207] John G. Moffat, Fabien Vincent, Jonathan A. Lee, Jörg Eder, and Marco Prunotto. "Opportunities and challenges in phenotypic drug discovery: an industry perspective". In: *Nature Reviews Drug Discovery* 16.8 (2017), pp. 531–543. ISSN: 1474-1784. DOI: 10.1038/nrd.2017.111. URL: https://doi.org/10.1038/nrd.2017.111.

[208] Jacques Louis Monod, Jeffries. Wyman, and Jean Pierre Changeux. "ON THE NATURE OF ALLOSTERIC TRANSITIONS: A PLAUSIBLE MODEL." In: *Journal of molecular biology* 12 (1965), pp. 88–118. URL: https://api.semanticscholar.org/CorpusID:14378.

[209] Jacqueline Montalibet and Brian P. Kennedy. "Using yeast to screen for inhibitors of protein tyrosine phosphatase 1B". In: *Biochemical Pharmacology* 68.9 (2004), pp. 1807–1814. ISSN: 0006-2952. DOI: https://doi.org/10.1016/j.bcp.2004.06.024. URL: https://www.sciencedirect.com/science/article/pii/S0006295204004575.

[210] H. L. Morgan. "The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service." In: *J. Chem. Doc.* 5.2 (May 1965), pp. 107–113. ISSN: 0021-9576. DOI: 10.1021/c160017a018. URL: https://doi.org/10.1021/c160017a018.

[211] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. "Critical assessment of methods of protein structure prediction (CASP) — round x". In: *Proteins: Structure, Function, and Bioinformatics* 82.S2 (2014), pp. 1–6. DOI: 10.1002/prot.24452.

[212] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. "Critical assessment of methods of protein structure prediction: Progress and new directions in round XI". In: *Proteins: Structure, Function, and Bioinformatics* 84.S1 (2016), pp. 4–14. DOI: 10.1002/prot.25064.

[213] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, and Anna Tramontano. "Critical assessment of methods of protein structure prediction (CASP)—round IX". In: *Proteins: Structure, Function, and Bioinformatics* 79.S10 (2011), pp. 1–5. DOI: 10.1002/prot.23200.

[214] Christopher W. Murray and David C. Rees. "The rise of fragment-based drug discovery". In: *Nature Chemistry* 1.3 (2009), pp. 187–192. ISSN: 1755-4349. DOI: 10.1038/nchem.217. URL: https://doi.org/10.1038/nchem.217.

[215] Alexey G. Murzin, Steven E. Brenner, Tim Hubbard, and Cyrus Chothia. "SCOP: A structural classification of proteins database for the investigation of sequences and structures". In: *Journal of Molecular Biology* 247.4 (1995), pp. 536–540. ISSN: 0022-2836. DOI: https://doi.org/10.1016/S0022-2836(05)80134-2. URL: https://www.sciencedirect.com/science/article/pii/S0022283605801342.

[216] Harini Narayanan, Fabian Dingfelder, Alessandro Butté, Nikolai Lorenzen, Michael Sokolov, and Paolo Arosio. "Machine Learning for Biologics: Opportunities for Protein Engineering, Developability, and Formulation". In: *Trends in Pharmacological Sciences* 42.3 (Mar. 2021), pp. 151–165. DOI: 10.1016/j.tips.2020.12.004. URL: https://doi.org/10.1016/j.tips.2020.12.004.

[217] Ali Nawaz, Shehroz S. Khan, and Amir Ahmad. "Ensemble of Autoencoders for Anomaly Detection in Biomedical Data: A Narrative Review". In: *IEEE Access* 12 (2024), pp. 17273–17289. DOI: 10.1109/ACCESS.2024.3360691.

[218] Vitalii Igorevich Nesterov, Mario Wieser, and Volker Roth. "3DMolNet: A Generative Network for Molecular Structures". In: *ArXiv* abs/2010.06477 (2020). URL: https://api.semanticscholar.org/CorpusID:222310243.

[219] Jeffrey J Nirschl et al. "A deep-learning classifier identifies patients with clinical heart failure using whole-slide images of H&E tissue". en. In: *PLoS One* 13.4 (Apr. 2018), e0192726. DOI: 10.1371/journal.pone.0192726.

[220] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. "A review on the attention mechanism of deep learning". In: *Neurocomputing* 452 (2021), pp. 48–62. URL: https://api.semanticscholar.org/CorpusID:233562906.

[221] OpenAI and et al. "GPT-4 Technical Report". In: (2024). arXiv: 2303.08774 [cs.CL].

[222] CA Orengo, AD Michie, S Jones, DT Jones, MB Swindells, and JM Thornton. "CATH – a hierarchic classification of protein domain structures". In: *Structure* 5.8 (1997), pp. 1093–1109. ISSN: 0969-2126. DOI: https://doi.org/10.1016/S0969-2126(97)00260-8. URL: https://www.sciencedirect.com/science/article/pii/S0969212697002608.

[223] Genevieve B. Orr and Klaus-Robert M¨uller, eds. *Neural Networks: Tricks of the Trade*. Springer Berlin Heidelberg, 1998. ISBN: 9783540494300. DOI: 10.1007/3-540-49430-8. URL: http://dx.doi.org/10.1007/3-540-49430-8.

[224] Daniel Osorio, Paola Rondón-Villarreal, and Rodrigo Torres. "The R Journal: Peptides: A Package for Data Mining of Antimicrobial Peptides". In: *The R Journal* 7 (1 2015). https://doi.org/10.32614/RJ-2015-001, pp. 4–14. ISSN: 2073-4859. DOI: 10.32614/RJ-2015-001.

[225] Si-sheng Ou-Yang, Jun-yan Lu, Xiang-qian Kong, Zhong-jie Liang, Cheng Luo, and Hualiang Jiang. "Computational drug discovery". In: *Acta Pharmacologica Sinica* 33.9 (2012), pp. 1131–1140. ISSN: 1745-7254. DOI: 10.1038/aps.2012.109. URL: https://doi.org/10.1038/aps.2012.109.

[226] Genady Paikin and Ayellet Tal. "Solving multiple square jigsaw puzzles with missing pieces". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 4832–4839. DOI: 10.1109/CVPR.2015.7299116.

[227] Gaia V. Paolini, Richard H. B. Shapland, Willem P. van Hoorn, Jonathan S. Mason, and Andrew L. Hopkins. "Global mapping of pharmacological space". In: *Nature Biotechnology* 24.7 (2006), pp. 805–815. ISSN: 1546-1696. DOI: 10.1038/nbt1228. URL: https://doi.org/10.1038/nbt1228.

[228] Steven M. Paul et al. "How to improve R&D productivity: the pharmaceutical industry's grand challenge". In: *Nature Reviews Drug Discovery* 9.3 (2010), pp. 203–214. ISSN: 1474-1784. DOI: 10.1038/nrd3078. URL: https://doi.org/10.1038/nrd3078.

[229] Marie-Morgane Paumard, David Picard, and Hedi Tabia. "Deepzzle: Solving Visual Jigsaw Puzzles With Deep Learning and Shortest Path Optimization". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 3569–3581. DOI: 10.1109/TIP.2019.2963378.

[230] RCSB PDB. *RCSB PDB*. URL: https://www.rcsb.org/ (visited on 09/15/2022).

[231] Eric F. Pettersen et al. "UCSF ChimeraX: Structure visualization for researchers, educators, and developers". In: *Protein Science* 30.1 (2021), pp. 70–82. DOI: https://doi.org/10.1002/pro.3943. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pro.3943. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/pro.3943.

[232] Alleyn T. Plowright, Craig Johnstone, Jan Kihlberg, Jonas Pettersson, Graeme Robb, and Richard A. Thompson. "Hypothesis driven drug design: improving quality and effectiveness of the design-make-test-analyse cycle." eng. In: *Drug discovery today* 17 (1-2 Jan. 2012), pp. 56–62. DOI: 10.1016/j.drudis.2011.09.012.

[233] Riccardo Poli, James Kennedy, and Tim Blackwell. "Particle swarm optimization". In: *Swarm Intelligence* 1.1 (2007), pp. 33–57. ISSN: 1935-3820. DOI: 10.1007/s11721-007-0002-0. URL: https://doi.org/10.1007/s11721-007-0002-0.

[234] Pavel G Polishchuk, Eugene N Muratov, Anatoly G Artemenko, Oleg G Kolumbin, Nail N Muratov, and Victor E Kuz'min. "Application of random forest approach to QSAR prediction of aquatic toxicity". In: *Journal of chemical information and modeling* 49.11 (Nov. 2009), 2481—2488. ISSN: 1549-9596. DOI: 10.1021/ci900203n. URL: https://doi.org/10.1021/ci900203n.

[235] Dolev Pomeranz, Michal Shemesh, and Ohad Ben-Shahar. "A fully automated greedy square jigsaw puzzle solver". In: *CVPR 2011*. IEEE, 2011, pp. 9–16. DOI: 10.1109/CVPR.2011.5995331.

[236] Katya B. Popova, Aikaterini Valsamatzi-Panagiotou, and Robert Penchovsky. "New drug discovery strategies for targeting drug-resistant bacteria". In: *Environmental Chemistry Letters* 19.3 (2021), pp. 1995–2004. ISSN: 1610-3661. DOI: 10.1007/s10311-021-01181-3. URL: https://doi.org/10.1007/s10311-021-01181-3.

[237] Lata Prasad, E Bruce Waygood, Jeremy S Lee, and Louis T J Delbaere. "The 2.5 Å resolution structure of the Jel42 Fab fragment/HPr complex". en. In: *J. Mol. Biol.* 280.5 (July 1998), pp. 829–845. DOI: 10.1006/jmbi.1998.1888.

[238] Prot pi. *Prot pi | Protein tool*. [Online; accessed 14-March-2023]. 2023. URL: https://www.protpi.ch/Calculator/ProteinTool.

[239] Protein Calculator v3.4. *Protein Calculator*. [Online; accessed 14-March-2023]. 2023. URL: https://protcalc.sourceforge.net/.

[240] Protein pKa calculations. *Protein pKa calculations — Wikipedia, The Free Encyclopedia*. [Online; accessed 14-March-2023]. 2023. URL: https://en.wikipedia.org/wiki/Protein_pKa_calculations\#cite_note-Hass2015-1.

[241] J. Ross Quinlan. *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN: 1558602380.

[242] Alec Radford and Karthik Narasimhan. "Improving Language Understanding by Generative Pre-Training". In: 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[243] Colin Raffel et al. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *J. Mach. Learn. Res.* 21.1 (Jan. 2020). ISSN: 1532-4435.

[244] Matthew Ragoza, Joshua Hochuli, Elisa Idrobo, Jocelyn Sunseri, and David Ryan Koes. "Protein–Ligand Scoring with Convolutional Neural Networks". In: *Journal of Chemical Information and Modeling* 57.4 (Apr. 2017), 942–957. ISSN: 1549-960X. DOI: 10.1021/acs.jcim.6b00740. URL: http://dx.doi.org/10.1021/acs.jcim.6b00740.

[245] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. *Hierarchical Text-Conditional Image Generation with CLIP Latents*. 2022. arXiv: 2204.06125 [cs.CV].

[246] Chakkodabylu S Ramesha. "How many leads from HTS? – Comment". In: *Drug Discovery Today* 5.2 (2000), pp. 43–44. ISSN: 1359-6446. DOI: https://doi.org/10.1016/S1359-6446(99)01444-0. URL: https://www.sciencedirect.com/science/article/pii/S1359644699014440.

[247] Marc' Aurelio Ranzato, Y-Lan Boureau, and Yann LeCun. "Sparse feature learning for deep belief networks". In: *Proceedings of the 20th International Conference on Neural Information Processing Systems*. NIPS'07. Vancouver, British Columbia, Canada: Curran Associates Inc., 2007, 1185–1192. ISBN: 9781605603520.

[248] Marc'Aurelio Ranzato, Christopher Poultney, Sumit Chopra, and Yann LeCun. "Efficient learning of sparse representations with an energy-based model". In: *Proceedings of the 19th International Conference on Neural Information Processing Systems*. NIPS'06. Canada: MIT Press, 2006, 1137–1144.

[249] Roshan Rao et al. "MSA Transformer". In: *bioRxiv* (2021). DOI: 10.1101/2021.02.12.430858. eprint: https://www.biorxiv.org/content/early/2021/02/13/2021.02.12.430858.full.pdf. URL: https://www.biorxiv.org/content/early/2021/02/13/2021.02.12.430858.

[250] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *Proceedings of the 31st International Conference on Machine Learning*.

Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, June 2014, pp. 1278–1286. URL: https://proceedings.mlr.press/v32/rezende14.html.

[251] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. "Contractive Auto-Encoders: Explicit Invariance During Feature Extraction". In: *International Conference on Machine Learning*. 2011. URL: https://api.semanticscholar.org/CorpusID:8141422.

[252] David Rogers and Mathew Hahn. "Extended-Connectivity Fingerprints". In: *J. Chem. Inf. Model.* 50.5 (May 2010), pp. 742–754. ISSN: 1549-9596. DOI: 10.1021/ci100050t. URL: https://doi.org/10.1021/ci100050t.

[253] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review* 65.6 (1958), 386–408. ISSN: 0033-295X. DOI: 10.1037/h0042519. URL: http://dx.doi.org/10.1037/h0042519.

[254] Burkhard Rost. "Twilight zone of protein sequence alignments". In: *Protein Engineering* 12.2 (1999). Cited by: 1285; All Open Access, Bronze Open Access, 85 – 94. DOI: 10.1093/protein/12.2.85. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0032962457&doi=10.1093%2fprotein%2f12.2.85&partnerID=40&md5=7e99578219dc0b6edb944107ae3a8ed0.

[255] M M Rubin and Jean Pierre Changeux. "On the nature of allosteric transitions: implications of non-exclusive ligand binding." In: *Journal of molecular biology* 21 2 (1966), pp. 265–74. URL: https://api.semanticscholar.org/CorpusID:29186201.

[256] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323 (1986), pp. 533–536. URL: https://api.semanticscholar.org/CorpusID:205001834.

[257] David E. Rumelhart and James L. Mcclelland. *Parallel distributed processing: explorations in the microstructure of cognition. Volume 1. Foundations*. Jan. 1986.

[258] Tariq Sadad, Raja Atif Aurangzeb, Mejdl Safran, Imran, Sultan Alfarhood, and Jungsuk Kim. "Classification of Highly Divergent Viruses from DNA/RNA Sequence Using Transformer-Based Models". In: *Biomedicines* 11.5 (2023). ISSN: 2227-9059. DOI: 10.3390/biomedicines11051323. URL: https://www.mdpi.com/2227-9059/11/5/1323.

[259] M.S. Sagiroglu and A. Ercil. "A Texture Based Matching Approach for Automated Assembly of Puzzles". In: *18th International Conference on Pattern Recognition (ICPR'06)*. Vol. 3. IEEE, 2006, pp. 1036–1041. DOI: 10.1109/ICPR.2006.184.

[260] Ekansh Sahu, Gautam Mishra, Hitesh Singh, and Vivek Kumar. "A Review on the Evolution of Jigsaw Puzzle Algorithms and the Way Forward". In: *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*. Institute of Electrical and Electronics Engineers (IEEE), 2022, pp. 281–287. DOI: 10.23919/INDIACom54597.2022.9763234.

[261] Seyedeh Zahra Sajadi, Mohammad Ali Zare Chahooki, Sajjad Gharaghani, and Karim Abbasi. "AutoDTI++: deep unsupervised learning for DTI prediction by autoencoders". In: *BMC Bioinformatics* 22.1 (2021), p. 204. ISSN: 1471-2105. DOI: 10.1186/s12859-021-04127-2. URL: https://doi.org/10.1186/s12859-021-04127-2.

[262] Ruslan Salakhutdinov and Geoffrey Hinton. "Semantic hashing". In: *International Journal of Approximate Reasoning* 50.7 (2009). Special Section on Graphical Models and Information Retrieval, pp. 969–978. ISSN: 0888-613X. DOI: https://doi.org/10.1016/j.ijar.2008.11.006. URL: https://www.sciencedirect.com/science/article/pii/S0888613X08001813.

[263] Soumitra Samanta, Steve O'Hagan, Neil Swainston, Timothy J. Roberts, and Douglas B. Kell. "VAE-Sim: A Novel Molecular Similarity Measure Based on a Variational Autoencoder". In: *Molecules* 25.15 (2020). ISSN: 1420-3049. DOI: 10.3390/molecules25153446. URL: https://www.mdpi.com/1420-3049/25/15/3446.

[264] A. L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229. DOI: 10.1147/rd.33.0210.

[265] Nicolae Sapoval et al. "Current progress and open challenges for applying deep learning across the biosciences". In: *Nature Communications* 13.1 (Apr. 2022). ISSN: 2041-1723. DOI: 10.1038/s41467-022-29268-7. URL: http://dx.doi.org/10.1038/s41467-022-29268-7.

[266] Jack W. Scannell, Alex Blanckley, Helen Boldon, and Brian Warrington. "Diagnosing the decline in pharmaceutical RD efficiency". In: *Nature Reviews Drug Discovery* 11.3 (2012), pp. 191–200. ISSN: 1474-1784. DOI: 10.1038/nrd3681. URL: https://doi.org/10.1038/nrd3681.

[267] Claire Schalon, Jean-Sébastien Surgand, Esther Kellenberger, and Didier Rognan. "A simple and fuzzy method to align and compare druggable ligand-binding sites". In: *Proteins Structure, Function, and Bioinformatics* 71.4 (2008), pp. 1755–1778. DOI: https://doi.org/10.1002/prot.21858. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.21858. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.21858.

[268] Michael Schlander, Karla Hernandez-Villafuerte, Chih-Yuan Cheng, Jorge Mestre-Ferrandiz, and Michael Baumann. "How Much Does It Cost to Research and

Develop a New Drug? A Systematic Review and Assessment". In: *PharmacoEconomics* 39.11 (2021), pp. 1243–1269. ISSN: 1179-2027. DOI: 10.1007/s40273-021-01065-y. URL: https://doi.org/10.1007/s40273-021-01065-y.

[269] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. "wav2vec: Unsupervised Pre-Training for Speech Recognition". In: Sept. 2019, pp. 3465–3469. DOI: 10.21437/Interspeech.2019-1873.

[270] Alexandra M. Schnoes, Shoshana D. Brown, Igor Dodevski, and Patricia C. Babbitt. "Annotation Error in Public Databases: Misannotation of Molecular Function in Enzyme Superfamilies". In: *PLOS Computational Biology* 5.12 (Dec. 2009), pp. 1–13. DOI: 10.1371/journal.pcbi.1000605. URL: https://doi.org/10.1371/journal.pcbi.1000605.

[271] Kyriakos Schwarz, Ahmed Allam, Nicolas Andres Perez Gonzalez, and Michael Krauthammer. "AttentionDDI: Siamese attention-based deep learning method for drug-drug interaction predictions." eng. In: *BMC bioinformatics* 22 (1 July 2021), p. 412. DOI: 10.1186/s12859-021-04325-y.

[272] Duncan E. Scott, Anthony G. Coyne, Sean A. Hudson, and Chris Abell. "Fragment-Based Approaches in Drug Discovery and Chemical Biology". In: *Biochemistry* 51.25 (June 2012), pp. 4990–5003. ISSN: 0006-2960. DOI: 10.1021/bi3005126. URL: https://doi.org/10.1021/bi3005126.

[273] David Sehnal et al. "Mol* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures". In: *Nucleic Acids Research* 49.W1 (May 2021), W431–W437. ISSN: 0305-1048. DOI: 10.1093/nar/gkab314. eprint: https://academic.oup.com/nar/article-pdf/49/W1/W431/38842088/gkab314.pdf. URL: https://doi.org/10.1093/nar/gkab314.

[274] Dennis J. Selkoe. "Folding proteins in fatal ways". In: *Nature* 426 (2003), pp. 900–904. DOI: 10.1038/nature02264.

[275] Claude E. Shannon. "A mathematical theory of communication". In: *Bell Syst. Tech. J.* 27 (1948), pp. 623–656. URL: https://api.semanticscholar.org/CorpusID:55379485.

[276] Chandra Shekhar. "In Silico Pharmacology: Computer-Aided Methods Could Transform Drug Development". In: *Chemistry & Biology* 15.5 (2008), pp. 413–414. ISSN: 1074-5521. DOI: https://doi.org/10.1016/j.chembiol.2008.05.001. URL: https://www.sciencedirect.com/science/article/pii/S1074552108001610.

[277] Dror Sholomon, Omid David, and Nathan S. Netanyahu. "A Genetic Algorithm-Based Solver for Very Large Jigsaw Puzzles". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. Institute of Electrical and Electronics Engineers (IEEE), 2013, pp. 1767–1774. DOI: 10.1109/CVPR.2013.231.

[278] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. "The Curse of Recursion: Training on Generated Data Makes Models Forget". In: *ArXiv* abs/2305.17493 (2023). URL: https://api.semanticscholar.org/CorpusID:258987240.

[279] Tong Ying Shun, John S. Lazo, Elizabeth R. Sharlow, and Paul A. Johnston. "Identifying Actives from HTS Data Sets: Practical Approaches for the Selection of an Appropriate HTS Data-Processing Method and Quality Control Review". In: *J Biomol Screen* 16.1 (Dec. 2010), pp. 1–14. ISSN: 1087-0571. DOI: 10.1177/1087057110389039. URL: https://doi.org/10.1177/1087057110389039.

[280] Sandy Lim Siew Mooi, Abu Bakar Md Sultan, Md Sulaiman, Aida Mustapha, and Kuan Yew Leong. "Crossover and Mutation Operators of Genetic Algorithms". In: *International Journal of Machine Learning and Computing* 7 (Feb. 2017), pp. 9–12. DOI: 10.18178/ijmlc.2017.7.1.611.

[281] David So, Quoc Le, and Chen Liang. "The Evolved Transformer". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 5877–5886. URL: https://proceedings.mlr.press/v97/so19a.html.

[282] *Step 3: Clinical Research — fda.gov*. https://www.fda.gov/patients/drug-development-process/step-3-clinical-research. [Accessed 04-05-2024].

[283] Nisan Stiennon et al. "Learning to summarize with human feedback". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 3008–3021. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

[284] Gary D. Stormo, Thomas D. Schneider, Larry Gold, and Andrzej Ehrenfeucht. " Use of the 'Perceptron' algorithm to distinguish translational initiation sites in E. coli". In: *Nucleic Acids Research* 10.9 (May 1982), pp. 2997–3011. ISSN: 0305-1048. DOI: 10.1093/nar/10.9.2997. eprint: https://academic.oup.com/nar/article-pdf/10/9/2997/7067690/10-9-2997.pdf. URL: https://doi.org/10.1093/nar/10.9.2997.

[285] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. "Segmenter: Transformer for Semantic Segmentation". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 7242–7252. DOI: 10.1109/ICCV48922.2021.00717.

[286] Dhingra Surbhi, Sowdhamini Ramanathan, Cadet Frederic, and Offmann Bernard. "A glance into the evolution of template-free protein structure prediction methodologies". In: *Biochimie* 175 (2020), pp. 85–92. DOI: 10.1016/j.biochi.2020.04.026.

[287] Vladimir Svetnik, Andy Liaw, Christopher Tong, J. Christopher Culberson, Robert P. Sheridan, and Bradley P. Feuston. "Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling". In: *Journal of Chemical Information and Computer Sciences* 43.6 (Nov. 2003), 1947–1958. ISSN: 0095-2338. DOI: 10.1021/ci034160g. URL: http://dx.doi.org/10.1021/ci034160g.

[288] David C. Swinney and Jason Anthony. "How were new medicines discovered?" In: *Nature Reviews Drug Discovery* 10.7 (2011), pp. 507–519. ISSN: 1474-1784. DOI: 10.1038/nrd3480. URL: https://doi.org/10.1038/nrd3480.

[289] Norberto Sánchez-Cruz, José L Medina-Franco, Jordi Mestres, and Xavier Barril. "Extended connectivity interaction features: improving binding affinity prediction through chemical description". In: *Bioinformatics* 37.10 (Dec. 2020), pp. 1376–1382. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btaa982. eprint: https://academic.oup.com/bioinformatics/article-pdf/37/10/1376/50360818/btaa982.pdf. URL: https://doi.org/10.1093/bioinformatics/btaa982.

[290] Alan Talevi. "Computer-Aided Drug Design: An Overview." eng. In: *Methods in molecular biology (Clifton, N.J.)* 1762 (2018), pp. 1–19. DOI: 10.1007/978-1-4939-7756-7_1.

[291] Jie Tan, John H. Hammond, Deborah A. Hogan, and Casey S. Greene. "ADAGE-Based Integration of Publicly Available Pseudomonas aeruginosa Gene Expression Data with Denoising Autoencoders Illuminates Microbe-Host Interactions." eng. In: *mSystems* 1 (1 2016).

[292] *The Nobel Prize in Chemistry 1990 — nobelprize.org*. https://www.nobelprize.org/prizes/chemistry/1990/corey/facts/. [Accessed 07-05-2024].

[293] *The Nobel Prize in Physics 1944 — nobelprize.org*. https://www.nobelprize.org/prizes/physics/1944/summary/. [Accessed 14-05-2024].

[294] *The Nobel Prize in Physics 1952 — nobelprize.org*. https://www.nobelprize.org/prizes/physics/1952/summary/. [Accessed 14-05-2024].

[295] Chuan-Kang Ting. "On the Mean Convergence Time of Multi-parent Genetic Algorithms Without Selection". In: *Advances in Artificial Life*. Ed. by Mathieu S. Capcarrère, Alex A. Freitas, Peter J. Bentley, Colin G. Johnson, and Jon Timmis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 403–412. ISBN: 978-3-540-31816-3.

[296] Stephen Todd et al. "CSynth: an interactive modelling and visualization tool for 3D chromatin structure". In: *Bioinformatics* 37.7 (Aug. 2020), pp. 951–955. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btaa757. eprint: https://academic.oup.com/bioinformatics/article-pdf/37/7/951/50341181/btaa757.pdf. URL: https://doi.org/10.1093/bioinformatics/btaa757.

[297] Hugo Touvron et al. "LLaMA: Open and Efficient Foundation Language Models". In: (2023). arXiv: 2302.13971 [cs.CL].

[298] Ludovic Trottier, Philippe Giguere, and Brahim Chaib-draa. "Parametric Exponential Linear Unit for Deep Convolutional Neural Networks". In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2017, pp. 207–214. DOI: 10.1109/ICMLA.2017.00038.

[299] A. M. TURING. "I.—COMPUTING MACHINERY AND INTELLIGENCE". In: *Mind* LIX.236 (Oct. 1950), pp. 433–460. ISSN: 0026-4423. DOI: 10.1093/mind/LIX.236.433. eprint: https://academic.oup.com/mind/article-pdf/LIX/236/433/30123314/lix-236-433.pdf. URL: https://doi.org/10.1093/mind/LIX.236.433.

[300] Anushka Tyagi, Aman Sharma, and Manish Bhardwaj. "Future of bioinformatics in India: A survey". In: *International journal of health sciences* 6.S2 (June 2022), 13767–13778. DOI: 10.53730/ijhs.v6nS2.8624. URL: https://sciencescholar.us/journal/index.php/ijhs/article/view/8624.

[301] Dr. Anantkumar Umbarkar and P. Sheth. "CROSSOVER OPERATORS IN GENETIC ALGORITHMS: A REVIEW". In: *ICTACT Journal on Soft Computing (Volume: 6 , Issue: 1)* 6 (Oct. 2015). DOI: 10.21917/ijsc.2015.0150.

[302] C. Devi Arockia Vanitha, D. Devaraj, and M. Venkatesulu. "Gene Expression Data Classification Using Support Vector Machine and Mutual Information-based Gene Selection". In: *Procedia Computer Science* 47 (2015). Graph Algorithms, High Performance Implementations and Its Applications (ICGHIA 2014), pp. 13–21. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.03.178. URL: https://www.sciencedirect.com/science/article/pii/S1877050915004469.

[303] Ashish Vaswani et al. "Attention is All You Need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA, 2017, 6000–6010.

[304] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. "Graph Attention Networks". In: (2018). DOI: https://doi.org/10.48550/arXiv.1710.10903. arXiv: 1710.10903 [stat.ML].

[305] C Venkateswaran, M Ramachandran, Ramu Kurinjimalar, Prasanth Vidhya, and G Mathivanan. "Application of Simulated Annealing in Various Field". In: *1* 1.1 (Feb. 2022), 01–08. DOI: 10.46632/mc/1/1/1. URL: http://dx.doi.org/10.46632/mc/1/1/1.

[306] Jorge A. Vila. "About the Protein Space Vastness". In: *The Protein Journal* 39.5 (Oct. 2020), pp. 472–475. DOI: 10.1007/s10930-020-09939-4. URL: https://doi.org/10.1007s10930-020-09939-4.

[307] Fabien Vincent, Arsenio Nueda, Jonathan Lee, Monica Schenone, Marco Prunotto, and Mark Mercola. "Phenotypic drug discovery: recent successes, lessons learned and new directions". In: *Nature Reviews Drug Discovery* 21.12 (2022), pp. 899–914. ISSN: 1474-1784. DOI: 10.1038/s41573-022-00472-w. URL: https://doi.org/10.1038/s41573-022-00472-w.

[308] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. "Extracting and composing robust features with denoising autoencoders". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: Association for Computing Machinery, 2008, 1096–1103. ISBN: 9781605582054. DOI: 10.1145/1390156.1390294. URL: https://doi.org/10.1145/1390156.1390294.

[309] Eric Voltà-Durán et al. "Controlling self-assembling and tumor cell-targeting of protein-only nanoparticles through modular protein engineering". In: *Science China Materials* 63.1 (Sept. 2019), pp. 147–156. DOI: 10.1007/s40843-019-9582-9. URL: https://doi.org/10.1007/s40843-019-9582-9.

[310] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. "A Comprehensive Survey of Loss Functions in Machine Learning". In: *Annals of Data Science* 9.2 (Apr. 2020), 187–212. ISSN: 2198-5812. DOI: 10.1007/s40745-020-00253-5. URL: http://dx.doi.org/10.1007/s40745-020-00253-5.

[311] ZhenZhou Wang and Adam Sobey. "A comparative review between Genetic Algorithm use in composite optimisation and the state-of-the-art in evolutionary computation". In: *Composite Structures* 233 (2020), p. 111739. ISSN: 0263-8223. DOI: https://doi.org/10.1016/j.compstruct.2019.111739. URL: https://www.sciencedirect.com/science/article/pii/S0263822319328636.

[312] Jason Wei et al. "Emergent Abilities of Large Language Models". In: *Transactions on Machine Learning Research* (2022). Survey Certification. ISSN: 2835-8856. DOI: https://doi.org/10.48550/arXiv.2206.07682. URL: https://openreview.net/forum?id=yzkSU5zdwD.

[313] David Weininger. "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules". In: *J. Chem. Inf. Comput. Sci.* 28.1 (Feb. 1988), pp. 31–36. ISSN: 0095-2338. DOI: 10.1021/ci00057a005. URL: https://doi.org/10.1021/ci00057a005.

[314] David Weininger, Arthur Weininger, and Joseph L. Weininger. "SMILES. 2. Algorithm for generation of unique SMILES notation". In: *J. Chem. Inf. Comput. Sci.* 29.2 (May 1989), pp. 97–101. ISSN: 0095-2338. DOI: 10.1021/ci00062a008. URL: https://doi.org/10.1021/ci00062a008.

[315] Edmund Whittaker. "Eddington's theory of the constants of Nature". In: *Math. Gaz.* 29.286 (1945), pp. 137–144.

[316] Peter Willett, John M. Barnard, and Geoffrey M. Downs. "Chemical Similarity Searching". In: *J. Chem. Inf. Comput. Sci.* 38.6 (Nov. 1998), pp. 983–996. ISSN: 0095-2338. DOI: 10.1021/ci9800211. URL: https://doi.org/10.1021/ci9800211.

[317] Zhenqin Wu et al. "MoleculeNet: a benchmark for molecular machine learning". In: *Chemical Science* 9.2 (2018), 513–530. ISSN: 2041-6539. DOI: 10.1039/c7sc02664a. URL: http://dx.doi.org/10.1039/C7SC02664A.

[318] Maciej Wójcikowski, Michał Kukiełka, Marta M. Stepniewska-Dziubinska, and Pawel Siedlecki. "Development of a protein-ligand extended connectivity (PLEC) fingerprint and its application for binding affinity predictions." eng. In: *Bioinformatics (Oxford, England)* 35 (8 Apr. 2019), pp. 1334–1341. DOI: 10.1093/bioinformatics/bty757.

[319] Chunqiu Xia, Shi-Hao Feng, Ying Xia, Xiaoyong Pan, and Hong-Bin Shen. "Fast protein structure comparison through effective representation learning with contrastive graph neural networks". In: *PLOS Computational Biology* 18.3 (Mar. 2022), pp. 1–21. DOI: 10.1371/journal.pcbi.1009986. URL: https://doi.org/10.1371/journal.pcbi.1009986.

[320] Hanguang Xiao, Li Li, Qiyuan Liu, Xiuhong Zhu, and Qihang Zhang. "Transformers in medical image segmentation: A review". In: *Biomedical Signal Processing and Control* 84 (2023), p. 104791. ISSN: 1746-8094. DOI: https://doi.org/10.1016/j.bspc.2023.104791. URL: https://www.sciencedirect.com/science/article/pii/S1746809423002240.

[321] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. "SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 12077–12090. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/64f1f27bf1b4ec22924fd0acb550c235-Paper.pdf.

[322] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. "How Powerful are Graph Neural Networks?" In: *ArXiv* abs/1810.00826 (2018). DOI: https://doi.org/10.48550/arXiv.1810.00826. URL: https://api.semanticscholar.org/CorpusID:52895589.

[323] Venkata V. B. Yallapragada, Sidney P. Walker, Ciaran Devoy, Stephen Buckley, Yensi Flores, and Mark Tangney. "Function2Form Bridge—Toward synthetic protein holistic performance prediction". In: *Proteins: Structure, Function, and Bioinformatics* 88.3 (2020), pp. 462–475. DOI: https://doi.org/10.1002/prot.25825. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.25825. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.25825.

[324] Jianyi Yang and Yang Zhang. "I-TASSER server: new development for protein structure and function predictions." eng. In: *Nucleic acids research* 43 (W1 July 2015), W174–81. DOI: 10.1093/nar/gkv342.

[325] Zhenyu Yang, Xiaoxi Zeng, Yi Zhao, and Runsheng Chen. "AlphaFold2 and its applications in the fields of biology and medicine". In: *Signal Transduction and Targeted Therapy* 8.1 (Mar. 2023). ISSN: 2059-3635. DOI: 10.1038/s41392-023-01381-z. URL: http://dx.doi.org/10.1038/s41392-023-01381-z.

[326] Shunyu Yao et al. "Tree of Thoughts: Deliberate Problem Solving with Large Language Models". In: *ArXiv* abs/2305.10601 (2023). DOI: https://doi.org/10.48550/arXiv.2305.10601. URL: https://api.semanticscholar.org/CorpusID:258762525.

[327] Jian Ye, Scott McGinnis, and Thomas L. Madden. "BLAST: improvements for better sequence analysis". In: *Nucleic Acids Research* 34.suppl_2 (July 2006), W6–W9. ISSN: 0305-1048. DOI: 10.1093/nar/gkl164. eprint: https://academic.oup.com/nar/article-pdf/34/suppl_2/W6/7622896/gkl164.pdf. URL: https://doi.org/10.1093/nar/gkl164.

[328] Ziqi Yin, Hao Wang, Kaito Horio, Daisuke Kawahara, and Satoshi Sekine. "Should We Respect LLMs? A Cross-Lingual Study on the Influence of Prompt Politeness on LLM Performance". In: (2024). DOI: https://doi.org/10.48550/arXiv.2402.14531. arXiv: 2402.14531 [cs.CL].

[329] Wenbo Yu, David J. Weber, and Alexander D. Jr MacKerell. "Computer-Aided Drug Design: An Update." eng. In: *Methods in molecular biology (Clifton, N.J.)* 2601 (2023), pp. 123–152. DOI: 10.1007/978-1-0716-2855-3_7.

[330] Shuguang Yuan, H.C. Stephen Chan, and Zhenquan Hu. "Using PyMOL as a platform for computational drug design". In: *WIREs Computational Molecular Science* 7.2 (2017), e1298. DOI: https://doi.org/10.1002/wcms.1298. eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wcms.1298. URL: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wcms.1298.

[331] Matthew D. Zeiler. *ADADELTA: An Adaptive Learning Rate Method*. 2012. DOI: 10.48550/ARXIV.1212.5701. URL: https://arxiv.org/abs/1212.5701.

[332] Daoan Zhang et al. "DNAGPT: A Generalized Pre-trained Tool for Versatile DNA Sequence Analysis Tasks". In: (2023). DOI: https://doi.org/10.48550/arXiv.2307.05628. arXiv: 2307.05628 [q-bio.GN].

[333] Jianping Zhang, Yizhan Huang, Weibin Wu, and Michael R. Lyu. "Transferable Adversarial Attacks on Vision Transformers with Token Gradient Regularization". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 16415–16424. DOI: 10.1109/CVPR52729.2023.01575.

[334] Pengchuan Zhang et al. "Multi-Scale Vision Longformer: A New Vision Transformer for High-Resolution Image Encoding". In: (2021). DOI: https://doi.org/10.48550/arXiv.2103.15358. arXiv: 2103.15358 [cs.CV].

[335] Wen Zhang, Yi Xiong, Meng Zhao, Hua Zou, Xinghuo Ye, and Juan Liu. "Prediction of conformational B-cell epitopes from 3D structures by random forests with a distance-based feature". In: *BMC Bioinformatics* 12.1 (Aug. 2011). ISSN: 1471-2105. DOI: 10.1186/1471-2105-12-341. URL: http://dx.doi.org/10.1186/1471-2105-12-341.

[336] Yang Zhang and Jeffrey Skolnick. "Scoring function for automated assessment of protein structure template quality". In: *Proteins* 57.4 (2004), pp. 702–710. DOI: 10.1002/prot.20264.

[337] Yang Zhang and Jeffrey Skolnick. "TM-align: a protein structure alignment algorithm based on the TM-score". In: *Nucleic Acids Research* 33.7 (Jan. 2005), pp. 2302–2309. ISSN: 0305-1048. DOI: 10.1093/nar/gki524. eprint: https://academic.oup.com/nar/article-pdf/33/7/2302/7127128/gki524.pdf. URL: https://doi.org/10.1093/nar/gki524.

[338] Liangzhen Zheng, Jingrong Fan, and Yuguang Mu. "OnionNet: a Multiple-Layer Intermolecular-Contact-Based Convolutional Neural Network for Protein-Ligand Binding Affinity Prediction". In: *ACS Omega* 4.14 (Oct. 2019), pp. 15956–15965. DOI: 10.1021/acsomega.9b01997. URL: https://doi.org/10.1021/acsomega.9b01997.

[339] Xiliang Zheng, LinFeng Gan, Erkang Wang, and Jin Wang. "Pocket-Based Drug Design: Exploring Pocket Space". In: *The AAPS Journal* 15.1 (2013), pp. 228–241. ISSN: 1550-7416. DOI: 10.1208/s12248-012-9426-6. URL: https://doi.org/10.1208/s12248-012-9426-6.

[340] Daquan Zhou et al. "DeepViT: Towards Deeper Vision Transformer". In: *ArXiv* abs/2103.11886 (2021). DOI: https://doi.org/10.48550/arXiv.2103.11886. URL: https://api.semanticscholar.org/CorpusID:232307695.

[341] Jie Zhou, Pei Ke, Xipeng Qiu, Minlie Huang, and Junping Zhang. "ChatGPT: potential, prospects, and limitations". In: *Frontiers of Information Technology & Electronic Engineering* 25.1 (2024), pp. 6–11. ISSN: 2095-9230. DOI: 10.1631/FITEE.2300089. URL: https://doi.org/10.1631/FITEE.2300089.

[342] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. "DNABERT-2: Efficient Foundation Model and Benchmark For Multi-Species Genome". In: (2024). DOI: https://doi.org/10.48550/arXiv.2306.15006. arXiv: 2306.15006 [q-bio.GN].

[343] Jinhua Zhu et al. "Incorporating BERT into Neural Machine Translation". In: *Eighth International Conference on Learning Representations (ICLR)*. Apr. 2020. URL: https://www.microsoft.com/en-us/research/publication/incorporating-bert-into-neural-machine-translation/.